وزارة التعليم العالي والبحث العلمي
جهاز الإشراف والتقويم العلمي
دائرة ضمان الجودة والاعتماد الأكاديمي
قسم الاعتماد

دليل وصف البرنامج الأكاديمي والمقرر الدراسي

2024

المقدمة:

يُعد البرنامج التعليمي بمثابة حزمة منسقة ومنظمة من المقررات الدراسية التي تشتمل على إجراءات وخبرات تنظم بشكل مفردات دراسية الغرض الأساس منها بناء وصقل مهارات الخريجين مما يجعلهم مؤهلين لتلبية متطلبات سوق العمل يتم مراجعته وتقييمه سنوياً عبر إجراءات وبرامج التدقيق الداخلي أو الخارجي مثل برنامج الممتحن الخارجي.

يقدم وصف البرنامج الأكاديمي ملخص موجز للسمات الرئيسة للبرنامج ومقرراته مبيناً المهارات التي يتم العمل على اكسابها للطلبة مبنية على وفق اهداف البرنامج الأكاديمي وتتجلى أهمية هذا الوصف لكونه يمثل الحجر الأساس في الحصول على الاعتماد البرامجي ويشترك في كتابته الملاكات التدريسية بإشراف اللجان العلمية في الأقسام العلمية.

ويتضمن هذا الدليل بنسخته الثانية وصفاً للبرنامج الأكاديمي بعد تحديث مفردات وفقرات الدليل السابق في ضوء مستجدات وتطورات النظام التعليمي في العراق والذي تضمن وصف البرنامج الأكاديمي بشكلها التقليدي نظام (سنوي، فصلي) فضلاً عن اعتماد وصف البرنامج الأكاديمي المعمم بموجب كتاب دائرة الدراسات ت م2906/3 في 2023/5/3 فيما يخص البرامج التي تعتمد مسار بولونيا أساساً لعملها.

وفي هذا المجال لا يسعنا إلا أن نؤكد على أهمية كتابة وصف البرامج الاكاديمية والمقررات الدراسية لضمان حسن سير العملية التعليمية.

مفاهيم ومصطلحات:

وصف البرنامج الأكاديمي: يوفر وصف البرنامج الأكاديمي ايجازاً مقتضباً لرؤيته ورسالته وأهدافه متضمناً وصفاً دقيقاً لمخرجات التعلم المستهدفة على وفق استراتيجيات تعلم محددة.

وصف المقرر: يوفر إيجازاً مقتضياً لأهم خصائص المقرر ومخرجات التعلم المتوقعة من الطالب تحقيقها مبرهناً عما إذا كان قد حقق الاستفادة القصوى من فرص التعلم المتاحة. ويكون مشتق من وصف البرنامج.

رؤية البرنامج: صورة طموحة لمستقبل البرنامج الأكاديمي ليكون برنامجاً متطوراً وملهماً ومحفزاً وواقعياً وقابلاً للتطبيق.

رسالة البرنامج: توضح الأهداف والأنشطة اللازمة لتحقيقها بشكل موجز كما يحدد مسارات تطور البرنامج واتجاهاته.

اهداف البرنامج: هي عبارات تصف ما ينوي البرنامج الأكاديمي تحقيقه خلال فترة زمنية محددة وتكون قابلة للقياس والملاحظة.

هيكلية المنهج: كافة المقررات الدراسية / المواد الدراسية التي يتضمنها البرنامج الأكاديمي على وفق نظام التعلم المعتمد (فصلي، سنوي، مسار بولونيا) سواء كانت متطلب (وزارة، جامعة، كلية وقسم علمي) مع عدد الوحدات الدراسية.

مخرجات التعلم: مجموعة متوافقة من المعارف والمهارات والقيم التي اكتسبها الطالب بعد انتهاء البرنامج الأكاديمي بنجاح ويجب أن يُحدد مخرجات التعلم لكل مقرر بالشكل الذي يحقق اهداف البرنامج.

استراتيجيات التعليم والتعلم: بأنها الاستراتيجيات المستخدمة من قبل عضو هيئة التدريس لتطوير تعليم وتعلم الطالب وهي خطط يتم إتباعها للوصول إلى أهداف التعلم. أي تصف جميع الأنشطة الصفية واللاصفية لتحقيق نتائج التعلم للبرنامج.

وزارة التعليم العالي والبحث العلمي
جـهاز الإشـراف والتقـويم العلـمي
دائرة ضمان الجودة والاعتماد الأكاديمي
قسم الاعتماد الدولي

استمارة وصف البرنامج الأكاديمي للكليات

للعام الدراسي 2024 – 2023

اسم الجامعة: الموصل
اسم الكلية: كلية علوم الحاسوب والرياضيات
القسم: الشبكات
تاريخ ملء الملف: 2024/4/1

اسم رئيس القسم	اسم مدير شعبة ضمان الجودة وتقويم الاداء	اسم معاون العميد للشؤون العلمية
أ.م. د.اياد حسين عبدالقادر	ا. م. د. محمد جاجان يونس	ا. د. صفوان عمر حسون
التاريخ 24 / 4 / 2024	التاريخ 24/ 4 /2024	التاريخ 24 / 4 / 2024
التوقيع	التوقيع	التوقيع

اسم عميد الكلية
أ. د. ضحى بشير عبد الله
التاريخ 24 / 4/ 2024
التوقيع

1. رؤية البرنامج

رؤيتنا هي أن نكون قوة رائدة في مجال الشبكات، وقيادة التقدم التكنولوجي والابتكار. نحن نسعى جاهدين لتهيئة بيئة تعليمية ديناميكية تغذي الإبداع والتفكير النقدي والتعاون. ومن خلال أحدث الأبحاث والشراكات الصناعية والتعلم التجريبي، نهدف إلى تزويد طلابنا بالمهارات والمعرفة اللازمة للتفوق في تصميم الشبكات وتكوينها وإدارتها وأمنها. تتمثل رؤيتنا في تخريج محترفين ذوي مهارات عالية وقابلين للتكيف ومتطلعين إلى الأمام وقادرين على مواجهة التحديات المعقدة لصناعة الشبكات. نحن نطمح إلى أن يتم الاعتراف بنا كشركة رائدة عالميًا في تعليم الشبكات، وإحداث تأثير إيجابي على المجتمع من خلال مساهمات خريجينا في تطوير تقنيات الشبكات

2. رسالة البرنامج

رسالتنا هي توفير التعليم والبحث عالي الجودة في مجال شبكات الكمبيوتر. نحن نهدف إلى تزويد طلابنا بالمعرفة والمهارات اللازمة لتصميم أنظمة الشبكات المعقدة وتكوينها وإدارتها وتأمينها، بالإضافة إلى تطوير تطبيقات مبتكرة وسهلة الاستخدام تستفيد من قوة الشبكات. ومن خلال المشاريع العملية والتدريب العملي والتعاون مع شركاء الصناعة، نقوم بتعزيز الإبداع والتفكير النقدي وقدرات حل المشكلات في سياق تطوير التطبيقات. مهمتنا هي تخريج محترفين ماهرين يمكنهم المساهمة في تطوير تقنيات الشبكات وإنشاء تطبيقات الويب والهواتف المحمولة المؤثرة التي تلبي احتياجات العالم الرقمي اليوم

3. اهداف البرنامج

يهدف قسم الشبكات إلى تحقيق أهداف البرنامج التالية:

1. المعرفة والمهارات: تزويد الطلاب بأساس قوي في مبادئ شبكات الكمبيوتر والبروتوكولات والتقنيات، مما يمكنهم من فهم وتحليل بنيات الشبكة وتكويناتها.
2. التصميم والتنفيذ: تزويد الطلاب بالقدرة على تصميم وتنفيذ شبكات كمبيوتر آمنة وفعالة، مع الأخذ في الاعتبار عوامل مثل قابلية التوسع والموثوقية والأداء.
3. إدارة الشبكة: تطوير المهارات في إدارة الشبكة وإدارتها، بما في ذلك مراقبة الشبكة واستكشاف الأخطاء وإصلاحها وتحسينها، لضمان التشغيل السلس وأداء البنى التحتية للشبكة.
4. تطوير التطبيقات: إعداد الطلاب للمساهمة في الطلب المتزايد على التطبيقات المتصلة بالشبكة، بما في ذلك تطبيقات الهاتف المحمول وتطبيقات الويب وحلول إنترنت الأشياء ، وتمكينهم من الاستفادة من الشبكات لتبادل البيانات بسلاسة وتعزيز

تجارب المستخدم.

5. الأمان: تعريف الطلاب بمفاهيم أمن الشبكات وأفضل الممارسات، وتمكينهم من تحديد نقاط الضعف والتهديدات في الشبكة والتخفيف منها، وتنفيذ تدابير أمنية فعالة.

6. التعاون والتواصل: تعزيز مهارات العمل الجماعي والتواصل والتعاون الفعال، مما يمكّن الطلاب من العمل بكفاءة في فرق متعددة التخصصات وتوصيل مفاهيم الشبكة المعقدة بشكل فعال إلى مختلف أصحاب المصلحة.

7. الاحتراف والممارسات الأخلاقية: غرس شعور قوي بالاحتراف والمسؤولية الأخلاقية لدى الطلاب، مع التركيز على أهمية النزاهة والخصوصية واتخاذ القرارات الأخلاقية في سياق تصميم الشبكات وإدارتها واستخدامها.

8. التعلم مدى الحياة: تعزيز شغف التعلم المستمر والتطوير المهني، وتشجيع الطلاب على البقاء على اطلاع دائم بتقنيات الشبكات الناشئة والتكيف مع المشهد المتطور لشبكات الكمبيوتر.

4. الاعتماد البرامجي

هل البرنامج حاصل على الاعتماد البرامجي ؟ ومن اي جهة ؟

Accreditation Board for Engineering and Technology (ABET)

5. المؤثرات الخارجية الأخرى

الامتحانات المركزية

6. هيكلية البرنامج

هيكل البرنامج	عدد المقررات	وحدة دراسية	النسبة المئوية	ملاحظات *
متطلبات المؤسسة	4	8	5.5%	
متطلبات الكلية	4	11	7.6%	
متطلبات القسم	45	125	86.8%	
التدريب الصيفي	–	–	–	
أخرى				

* ممكن ان تتضمن الملاحظات فيما اذا كان المقرر أساسي او اختياري .

7. وصف البرنامج

1st Stage

Semester 1 | 30 ECTS | 1 ECTS = 25 hrs

Code	Module	SSWL	USSWL	ECTS	Type	Pre-request

NT101	Information Technology Basics	78	72	6.00	B	None
NT102	Problems Solving & Programming 1	108	92	8.00	C	None
NT103	Calculus	48	77	5.00	S	None
NT104	Logic Design Fundamentals	93	82	7.00	C	None
UOM102	English Language 1	32	18	2.00	B	None
UOM104	Democracy and Human Rights	32	18	2.00	B	None

Semester 2 | 30 ECTS | 1 ECTS = 25 hrs

Code	Module	SSWL	USSWL	ECTS	Type	Pre-request
NT107	Problems Solving & Programming 2	78	72	6.00	C	NT102
NT108	Computer Organization	63	72	6.00	C	NT104
NT109	Data Communication & Networking	63	62	5.00	C	NT101
NT110	Probabilities & statistics	63	62	5.00	S	None
NT111	Discrete Mathematics	48	52	4.00	B	None
UOM101	Arabic Language	32	18	1.00	B	None
UOM103	Computer	48	27	3.00	B	None

2nd Stage

Second Year-*First Semester*
المرحلة الثانية- الفصل الاول

Subject	Theoretical	Practical	Units
Data Structures **هياكل البيانات**	2	2	3
Object Oriented Programming II **البرمجة الكيانية II**	2	2	3
Website Design &Programming I **تصميم وبرمجة المواقع I**	2	2	3
Network Protocols I **بروتوكولات الشبكة I**	3	-	3
Microprocessors & Interfacing I **المعالجات الدقيقة وأجهزة التواصل I**	2	2	3
Software Engineering **هندسة البرمجيات**	2	-	2
Total units **مجموع الوحدات**			17

Second Year- Second Semester

المرحلة الثانية- الفصل الثاني

Subject	Theoretical	Practical	Units
Introduction to Routing & Switching **مقدمة الى التوجيه والتبديل**	2	2	3
Visual Programming **البرمجة المرئية**	2	2	3
Computer Architecture **معمارية الحاسوب**	3	0	3
Algorithms Analysis & design **تحليل وتصميم الخوارزميات**	2	2	3
Principles of Data Bases **مبادئ قواعد البيانات**	2	2	3
Cybersecurity Principles **مبادئ الامن السبراني**	2	-	2
Website Design & Programming II **تصميم وبرمجة المواقع II**	2	2	3
Total Units			20

8. مخرجات التعلم المتوقعة للبرنامج	
المعرفة	
1. تشمل مخرجات التعلم لقسم شبكات الحاسوب ما يلي: 2. الفهم الشامل: تطوير فهم عميق لمبادئ شبكات الكمبيوتر والبروتوكولات والبنيات والتقنيات. 3. تصميم الشبكات وتنفيذها: اكتساب المهارات اللازمة لتصميم وتكوين وتنفيذ شبكات كمبيوتر آمنة وفعالة، مع الأخذ في الاعتبار عوامل مثل قابلية التوسع والموثوقية والأداء. 4. إدارة الشبكة وإدارتها: اكتساب الكفاءة في إدارة الشبكة ومراقبتها واستكشاف الأخطاء وإصلاحها وتحسينها لضمان التشغيل السلس وأداء البنى التحتية للشبكة. 5. أمن الشبكات: إظهار المعرفة بمفاهيم أمن الشبكات وأفضل الممارسات، وتطبيق التدابير الفعالة لحماية الشبكات من نقاط الضعف والتهديدات. 6. تطوير التطبيقات: تطوير القدرة على تصميم وتنفيذ تطبيقات الشبكات والهواتف الذكية والتطبيقات المستندة إلى الويب التي تستفيد من شبكات الكمبيوتر لتعزيز تجارب المستخدم وتلبية متطلبات العالم الحقيقي. 7. التعاون والتواصل: تعزيز مهارات العمل الجماعي والتواصل	1. نظرية 2. عملية 3. تدريب الطالب / تدريب صيفي 4. بحث تخرج

والتعاون الفعالة اللازمة للعمل في فرق متعددة التخصصات وتوصيل مفاهيم الشبكة بشكل فعال إلى مختلف أصحاب المصلحة. 8. الاحتراف والممارسات الأخلاقية: الالتزام بالمعايير المهنية والأخلاقية في تصميم الشبكات وإدارتها واستخدامها، وإظهار النزاهة واتخاذ القرارات المسؤولة. 9. التعلم مدى الحياة: تنمية عقلية التعلم المستمر والتطوير المهني للتكيف مع تقنيات الشبكات الناشئة واتجاهات الصناعة المتطورة.	
المهارات	
1. مهارة حل المشاكل وبرمجتها. 2. مهارة تهيئة وتصميم الشبكات المحلية. 3. مهارة بناء تطبيقات الشبكات. 4. مهارة برمجة التطبيقات القائمة على الويب. 5. مهارة المناقشة واتخاذ القرارات الصائبة. 6. مهارة استخدام الوسائل الحديثة منها الحاسوب. 7. مهارة البحث عن المعلومة العلمية الصحيحة. 8. مهارة اجراء البحوث العلمية وتحليلها وحل المشاكل الخاصة بها ووضع الاستنتاجات المناسبة في حلها لغرض اتخاذ القرار.	1. القدرة على الدراسة الجماعية. 2. القدرة على المناقشة العلمية بين الطلبة. 3. القدرة في تنمية المهارات بين الطلبة. 4. القدرة في المناقشة والتحليل واتخاذ القرار الجماعي. تنمية القدرة على التعاون.
القيم	
• إظهار الوعي بالقضايا الأخلاقية المتعلقة بخصوصية البيانات والسرية والملكية الفكرية. • الالتزام بالمبادئ التوجيهية الأخلاقية المتعلقة بالشبكات وامنيتها. • احتضان التعلم مدى الحياة والبقاء على اطلاع دائم بالاتجاهات والتقنيات الناشئة في هذا المجال.	

9. استراتيجيات التعليم والتعلم

1. التطلع المستمر نحو التفوق المعرفي في التعليم والبحث العلمي والخدمة الاحترافية في مختلف العلوم.
2. إعداد الطلبة لسوق العمل وتنمية قدراتهم على التفاعل والتواصل مع الآخرين من خلال المشاركة الفعالة في برنامج التدريب الميداني.
3. اكتساب المهارات لعرض الأفكار والعمل ضمن فريق واحد وذلك من خلال مشاريع التخرج.
4. تأهيل الطلبة للدراسات العليا في مجال شبكات الحاسوب.
5. إعداد القيادات العلمية المتخصصة من خلال برنامج الدراسات العليا.
6. التفاعل مع العلوم الأخرى وبالأخص منها الاحصاء والرياضيات والحاسوب.

10. طرائق التقييم

1. الامتحانات الالكترونية (On line).
2. الامتحانات المركزية والشهرية.
3. الامتحانات اليومية.
4. الواجبات اليومية.
5. التقارير العلمية
6. الامتحانات المختبرية الحاسوبية.
7. مشاريع التخرج.

11. الهيئة التدريسية

أعضاء هيئة التدريس

الرتبة العلمية	التخصص		المتطلبات/المهارات الخاصة (ان وجدت)		اعداد الهيئة التدريسية	
	عام	خاص			ملاك	محاضر
استاذ مساعد	علوم الحاسوب	شبكات الحاسوب			2	
مدرس	علوم الحاسوب	شبكات الحاسوب			2	
مدرس	علوم الحاسوب	شبكات الحاسوب			1	
مدرس	علوم الحاسوب	امنية وادارة الشبكات			1	
مدرس	علوم الحاسوب	هندسة البرمجيات			1	
مدرس	علوم الحاسوب	شبكات معقدة			1	
مدرس	علوم الحاسوب	ذكاء اصطناعي			1	
مدرس مساعد	علوم الحاسوب	علوم الحاسوب			3	
مدرس مساعد	قانون	قانون			1	

التطوير المهني

توجيه أعضاء هيئة التدريس الجدد

1. التعليم الالكتروني.
2. استخدام وسائل الانترنيت.
3. استخدام وسائل التواصل الحديثة.
4. استخدام وسائل الاتصال الحديثة.
5. نشاطات لا صفية.
6. دورات تدريبية متطورة في تعلم البرامج الحديثة.
7. الاستشارات العلمية وسبل تطورها وتطبيقها في مختلف المجالات.

التطوير المهني لأعضاء هيئة التدريس

تصف بإيجاز خطة وترتيبات التطوير الأكاديمي والمهني لأعضاء هيئة التدريس كاستراتيجيات التدريس والتعلم، وتقييم نتائج التعلم، التطوير المهني وما الى ذلك.

12. معيار القبول

1. القبول المركزي في وزارة التعليم العالي والبحث العلمي.
2. معدل الطالب ضمن قوائم القبول المركزي باستثناء ابناء التدريسيين وبناء الشهداء والامتيازات التي تنص عليها التعليمات الخاصة بالوزارة حيث يتم قبولهم حسب الرغبة لتوزيعهم على الاقسام العلمية.

13. أهم مصادر المعلومات عن البرنامج

14. خطة تطوير البرنامج

- تطوير منهج منقح يعكس اتجاهات الصناعة الحالية والتقنيات وأفضل الممارسات.
- دمج المفاهيم الأساسية مع المواضيع المتقدمة مثل الحوسبة السحابية، والأمن السيبراني، وإنترنت الأشياء.
- دمج المعامل العملية والمشاريع ودراسات الحالة لتعزيز المهارات العملية وقدرات حل المشكلات.

مخطط مهارات المنهج

السنة / المستوى	رمز المقرر	اسم المقرر	اساسي ام اختياري	مخرجات التعلم المطلوبة من البرنامج											
				المعرفة				المهارات				القيم			
				أ1	أ2	أ3	أ4	ب1	ب2	ب3	ب4	ج1	ج2	ج3	ج4
السنة الأول/ الفصل الاول	NT101	اساسيات تكنولوجيا المعلومات	اساسي	√	√	√	√	√	√	√	√	√	√	√	
	NT102	حل المشاكل وبرمجتها 1	اساسي	√	√	√	√		√	√	√		√	√	
	NT103	حساب التفاضل والتكامل	اساسي		√	√	√	√	√	√	√	√	√	√	√
	NT104	أساسيات التصميم المنطقي	اساسي	√	√	√	√	√	√	√	√	√	√	√	√
	UOM102	اللغة الانكليزية 1	اساسي		√			√					√	√	√
	UOM104	الديمقراطية وحقوق الانسان	اساسي		√	√		√	√	√	√		√	√	√
السنة الأول/ الفصل الثاني	NT107	حل المشاكل وبرمجتها 2	اساسي	√				√		√			√		
	NT108	تنظيم الحاسوب	اساسي		√				√	√			√		
	NT109	اتصالات البيانات والشبكات	اساسي	√	√	√		√	√	√	√	√	√	√	√
	NT110	الاحتمالات والإحصاء	اساسي	√	√	√		√	√	√	√	√	√	√	√
	NT111	الرياضيات المتقطعة	اساسي	√					√	√	√	√	√	√	√
	UOM101	اللغة العربية	اساسي	√	√	√		√	√	√	√		√	√	√
	UOM103	الحاسوب	اساسي												

السنة / المستوى	رمز المقرر	اسم المقرر	اساسي ام اختياري	المعرفة								المهارات				القيم			
				أ1	أ2	أ3	أ4	ب1	ب2	ب3	ب4	ج1	ج2	ج3	ج4
السنة الثانية/ الفصل الاول	NET201	هياكل البيانات	اساسي	√				√	√	√	√	√		√	√
	NET202	البرمجة الكيانية 2	اساسي	√	√	√	√	√	√	√	√	√	√	√	√
	NET203	تصميم وبرمجة المواقع 1	اساسي		√	√	√		√	√	√		√	√	√
	NET204	بروتوكولات الشبكة1	اساسي	√	√	√	√		√	√	√		√	√	√
	NET205	المعالجات الدقيقة واجهزة التواصل	اساسي	√	√	√	√		√			√		√	√
	NET206	هندسة البرمجيات	اساسي						√				√		
	NET207	جرائم البعث	اساسي												
السنة الثانية/ الفصل الثاني	NET208	مقدمة الى التوجيه والتبديل	اساسي		√	√		√		√	√		√		
	NET209	البرمجة المرئية	اساسي	√	√	√		√		√	√	√	√	√	√
	NET210	معمارية الحاسوب	اساسي	√	√	√		√	√	√	√	√	√	√	√
	NET211	تحليل وتصميم الخوارزميات	اساسي		√				√			√			
	NET212	مبادئ قواعد البيانات	اساسي	√				√					√		
	NET2013	مبادئ الامن السبراني	اساسي	√		√	√	√	√	√		√		√	√
	NET214	تصميم وبرمجة المواقع 2	اساسي	√		√	√	√	√	√		√		√	√
	NET215	اللغة الانكليزية 2	اساسي	√		√	√	√	√					√	√

MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information

معلومات المادة الدراسية

Module Title	**Information Technology Basics**	**Module Delivery**	
Module Type	**Basic**	☒ **Theory**	
Module Code	**NT101**	☒ **Lecture**	
ECTS Credits	**6**	☒ **Lab**	
SWL (hr/sem)	**150**	☐ **Tutorial**	
		☐ **Practical**	
		☐ **Seminar**	

Module Level	1	**Semester of Delivery**	1
Administering Department	NT	**College**	CSM
Module Leader	Name	**e-mail**	E-mail
Module Leader's Acad. Title	Professor	**Module Leader's Qualification**	Ph.D.
Module Tutor	Name (if available)	**e-mail**	E-mail
Peer Reviewer Name	Name	**e-mail**	E-mail
Scientific Committee Approval Date	18/06/2023	**Version Number**	1.0

Relation with other Modules

العلاقة مع المواد الدراسية الأخرى

Prerequisite module	None	**Semester**	
Co-requisites module	None	**Semester**	

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	This course introduces students to the essential technical and professional skills required in the field of Information Technology (IT). Through written assignments, students gain an understanding of the operation of computers, computer networks, Internet fundamentals, programming, and computer support. We hope the students also learn about the social impact of technological change and the ethical issues related to technology. Throughout the course, instructional activities emphasize safety, professionalism, accountability, and eefficiency for workers within the eld of IT. Indeed, this course of an Information Technology under Network Department cover the infrastructure model of information technology discipline that deals with the computation of hardware and software, involving other programming languages, which form an essential part of Information Technology. Therefore, the course offers overview in various elds for information technology in terms of computer networks such as Data Science, Cloud Computing, Software Engineering, and also Artificial Intelligence, Machine Learning, Block-chain Engineering, etc. 1. looking forward to taking up a profession in Information Technology can take this course. Comprehension of Computer Systems: Understand the components of a computer system, including hardware and software, and how they interact to perform tasks. 2. Proficiency in Operating Systems: Install, configure, and manage operating systems, perform file management operations, and utilize basic system administration tasks. 3. Knowledge of Networks and Connectivity: Explain network architectures, protocols, and technologies, and demonstrate the ability to set up basic networks, configure IP addresses, and understand data transmission concepts. 4. Practical Skills Application: Utilize hands-on exercises and practical assignments to apply theoretical concepts, including computer hardware installation, software setup, network configuration, and basic programming tasks. 5. Understanding of Professional Ethics and Communication: Identify ethical considerations in the IT field, including privacy, intellectual property, and responsible technology use. Exhibit effective communication skills for collaboration and documentation purposes.
Indicative Contents المحتويات الإرشادية	The indicative contents of the IT basics course for the computer department may include the following topics: 1. Introduction to Information Technology: o Definition and scope of information technology

- Evolution and impact of IT on society and businesses
- Ethical considerations and challenges in IT

2. Computer Systems and Hardware:
   - Components of a computer system (CPU, memory, storage, input/output devices)
   - Computer organization and architecture
   - Digital data representation (binary, hexadecimal)
3. Operating Systems:
   - Functions and types of operating systems (e.g., Windows, macOS, Linux)
   - Process management and multitasking
   - Memory management and virtual memory
   - File systems and file management
4. Software and Applications Development:
   - Programming languages and their characteristics (e.g., Java, Python, C++)
   - Algorithms and problem-solving techniques
   - Introduction to software development methodologies (e.g., waterfall, agile)
5. Networking Fundamentals:
   - Network topologies (e.g., bus, star, mesh)
   - Network protocols (e.g., TCP/IP, HTTP, DNS)
   - Local area networks (LANs) and wide area networks (WANs)
   - Network security and basic concepts of cybersecurity
6. Database Systems:
   - Introduction to database management systems (DBMS)
   - Relational database concepts and design principles
   - Structured Query Language (SQL) for data manipulation and retrieval
   - Basic database administration tasks and data integrity
7. Web Technologies:
   - Basics of web development (HTML, CSS, JavaScript)

	○ Client-server architecture and web application deployment ○ Web design principles and usability considerations ○ Introduction to content management systems (CMS) 8. Information Security: ○ Fundamentals of information security and data protection ○ Common security threats and vulnerabilities ○ Authentication and access control mechanisms ○ Encryption and cryptographic techniques 9. Emerging Technologies: ○ Cloud computing and virtualization ○ Artificial intelligence and machine learning ○ Big data analytics and data science ○ Internet of Things (IoT) and its applications 10. Ethical and Legal Considerations: ○ Ethical issues in IT, including privacy and responsible use of technology ○ Intellectual property rights and copyright laws ○ Cybersecurity laws and regulations ○ Data protection and privacy laws (e.g., GDPR, CCPA)

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	Learning and teaching strategies for the IT basics course for the Network department can include a combination of the following: 1. Active Learning: Incorporate activities that engage students actively in the learning process. This can include hands-on exercises, group projects, case studies, and discussions. Encourage students to apply the concepts they learn to real-world scenarios and problems.

2. Practical Exercises: Provide opportunities for students to practice and apply their knowledge. Assign programming exercises, database design projects, or networking simulations to reinforce understanding and develop practical skills. Offer guidance and feedback during the exercises to facilitate learning.
3. Real-World Examples: Use relevant and relatable examples from various industries and domains to illustrate the application of IT concepts. Demonstrate how technology is used in business processes, healthcare, finance, or other fields to make the concepts more tangible and meaningful.
4. Multimedia Resources: Supplement lectures and readings with multimedia resources such as videos, interactive tutorials, online demonstrations, and virtual labs. Visual and interactive content can help students grasp complex concepts and engage different learning styles.
5. Guest Speakers and Industry Connections: Invite guest speakers from the IT industry or academia to share their experiences, insights, and real-world applications of IT concepts. Establish connections with professionals working in the field to provide students with networking opportunities and industry perspectives.
6. Problem-Solving and Critical Thinking: Emphasize problem-solving and critical-thinking skills throughout the course. Encourage students to analyze and evaluate information, think creatively, and propose solutions to IT-related challenges. Pose thought-provoking questions and scenarios to stimulate discussion and higher-level thinking.
7. Assessment Variety: Use a variety of assessment methods to evaluate students' understanding and mastery of the concepts. This can include quizzes, exams, projects, presentations, and portfolios. Incorporate both individual and group assessments to encourage collaboration and teamwork skills.
8. Current and Emerging Trends: Introduce students to current trends and emerging technologies in the field of information technology. Discuss their impact, potential applications, and challenges. Encourage students to explore and research these topics to stay updated with the evolving IT landscape.
9. Ethical and Legal Considerations: Incorporate discussions and assignments that focus on ethical and legal issues in IT. Promote discussions on responsible use of technology, privacy concerns, intellectual property rights, and cybersecurity ethics. Foster awareness of the ethical implications of IT in society.
10. Continuous Feedback and Support: Provide regular feedback on students' progress, both during in-class activities and through assignments. Offer opportunities for students to seek clarification and ask questions. Create a supportive learning environment that encourages

	open communication and collaboration. These strategies promote active learning, practical application of knowledge, and engagement with the subject matter. They cater to different learning styles and encourage students to develop critical thinking, problem-solving, and communication skills necessary for success in the IT field.

Student Workload (SWL)

الحمل الدراسي للطالب محسوب لـ ١٥ اسبوعا

Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	63	**Structured SWL (h/w)** الحمل الدراسي المنتظم للطالب أسبوعيا	6
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	62	**Unstructured SWL (h/w)** الحمل الدراسي غير المنتظم للطالب أسبوعيا	6
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	125		

Module Evaluation

تقييم المادة الدراسية

		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	**Quizzes**	2	10% (10)	5 and 10	LO #1, #2 and #10, #11
	Assignments	2	10% (10)	2 and 12	LO #3, #4 and #6, #7
	Projects / Lab.	1	10% (10)	Continuous	All
	Report	1	10% (10)	13	LO #5, #8 and #10
Summative	**Midterm Exam**	2hr	10% (10)	7	LO #1 - #7

assessment	**Final Exam**	3hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus) المنهاج الاسبوعي النظري	
	Material Covered
Week 1	Introduction to the basic concepts of Information Technology (IT) and their applications.
Week 2	Understand the structure layers of the infrastructure model of Information technology environment, especially about end-users, operating system, computer network, and storage.
Week 3	Computer Hardware • Components of a computer system • Input and output devices • Computer peripherals and their functions
Week 4	Computer Software • Types of software: system software and application software • Operating systems and their features • Software installation and management
Week 4	Practical Skills - Operating Systems Basic concepts under IT infrastructure model layers • Installation and configuration of operating systems • File management operations • System administration tasks
Week 5	End-users concept underlying infrastructure model
Week 6	Networks and Connectivity-Basic concepts under IT infrastructure model layers • Network architectures: LAN, WAN, WLAN • Network protocols: TCP/IP, Ethernet, Wi-Fi
Week 7	Networks and Connectivity Basic concepts under IT infrastructure model layers • Network architectures: LAN, WAN, WLAN

	• Network protocols: TCP/IP, Ethernet, Wi-Fi
Week 8	Mid term Examination
Week 9	Storage concept underlying infrastructure model
Week 10	Introduction to Data Management Basic concepts under IT infrastructure model layers • Basics of databases and data management systems
Week 11	Introduction to Data Management Basic concepts under IT infrastructure model layers • Structured Query Language (SQL)
Week 12	Introduction to Web Development Basic concepts under IT infrastructure model layers • HTML and CSS fundamentals
Week 13	Introduction to Web Development Basic concepts under IT infrastructure layers • Web page creation and design principles
Week 14	Ethical Considerations Basic concepts under IT infrastructure layers • Ethics in the IT field: privacy, intellectual property, responsible technology use • Professional communication and documentation skills
Week 15	Week 15: Review • Review of key concepts covered throughout the course • Completion of final projects or assignments demonstrating understanding of IT basics
Week 16	**Preparatory week before the final Exam**

Delivery Plan (Weekly Lab. Syllabus) المنهاج الأسبوعي للمختبر	
	Material Covered
Week 1	Operating System Installation and Configuration o Objective: Enable students to install and configure an operating system. o Activities: Students will install an operating system of their choice (e.g., Windows, Linux) on a virtual machine or physical computer. They will configure settings, create user accounts, and explore basic system

	administration tasks.
Week 2	Introduction to use the terminal of Ubuntu Operating System
Week 3	Introduction to Ubuntu environment.
Week 4	Introduction to the Shell
Week 5	Navigation
Week 6	Navigation
Week 7	Exploring the System
Week 8	Mid term Exanimation
Week 9	Manipulating Files and Directories
Week 10	Permissions
Week 11	Processes
Week 12	Configuration and the Environment
Week 13	Storage Media
Week 14	Networking
Week 15	Networking

Learning and Teaching Resources

مصادر التعلم والتدريس

	Text	Available in the Library?
Required Texts		No
Recommended Texts		
Websites		

Grading Scheme

مخطط الدرجات				
Group	**Grade**	التقدير	**Marks %**	**Definition**
Success Group (50 - 100)	**A -** Excellent	امتياز	90 – 100	Outstanding Performance
	B - Very Good	جيد جدا	80 – 89	Above average with some errors
	C - Good	جيد	70 – 79	Sound work with notable errors
	D - Satisfactory	متوسط	60 – 69	Fair but with major shortcomings
	E - Sufficient	مقبول	50 – 59	Work meets minimum criteria
Fail Group (0 – 49)	**FX –** Fail	راسب (قيد المعالجة)	(45-49)	More work required but credit awarded
	F – Fail	راسب	(0-44)	Considerable amount of work required

Note: Marks Decimal places above or below 0.5 will be rounded to the higher or lower full mark (for example a mark of 54.5 will be rounded to 55, whereas a mark of 54.4 will be rounded to 54. The University has a policy NOT to condone "near-pass fails" so the only adjustment to marks awarded by the original marker(s) will be the automatic rounding outlined above.

MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information

معلومات المادة الدراسية

Module Title	**Information Technology Basics**	**Module Delivery**	
Module Type	**Basic**	☒ **Theory**	
Module Code	**NT101**	☒ **Lecture**	
ECTS Credits	**6**	☒ **Lab**	
SWL (hr/sem)	**150**	☐ **Tutorial**	
		☐ **Practical**	
		☐ **Seminar**	

Module Level	1	**Semester of Delivery**	1
Administering Department	NT	**College**	CSM
Module Leader	Name	**e-mail**	E-mail

Module Leader's Acad. Title	Professor	**Module Leader's Qualification**	Ph.D.
Module Tutor	Name (if available)	**e-mail**	E-mail
Peer Reviewer Name	Name	**e-mail**	E-mail
Scientific Committee Approval Date	18/06/2023	**Version Number**	1.0

Relation with other Modules

العلاقة مع المواد الدراسية الأخرى

Prerequisite module	None	**Semester**	
Co-requisites module	None	**Semester**	

Module Aims, Learning Outcomes and Indicative Contents

أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية

Module Learning Outcomes مخرجات التعلم للمادة الدراسية	This course introduces students to the essential technical and professional skills required in the field of Information Technology (IT). Through written assignments, students gain an understanding of the operation of computers, computer networks, Internet fundamentals, programming, and computer support. We hope the students also learn about the social impact of technological change and the ethical issues related to technology. Throughout the course, instructional activities emphasize safety, professionalism, accountability, and eefficiency for workers within the eld of IT. Indeed, this course of an Information Technology under Network Department cover the infrastructure model of information technology discipline that deals with the computation of hardware and software, involving other programming languages, which form an essential part of Information Technology. Therefore, the course offers overview in various elds for information technology in terms of computer networks such as Data Science, Cloud Computing, Software Engineering, and also Artificial Intelligence, Machine Learning, Block-chain Engineering, etc. 2. looking forward to taking up a profession in Information Technology can take this course. Comprehension of Computer Systems: Understand the components of a computer system, including hardware and

	software, and how they interact to perform tasks. 6. Proficiency in Operating Systems: Install, configure, and manage operating systems, perform file management operations, and utilize basic system administration tasks. 7. Knowledge of Networks and Connectivity: Explain network architectures, protocols, and technologies, and demonstrate the ability to set up basic networks, configure IP addresses, and understand data transmission concepts. 8. Practical Skills Application: Utilize hands-on exercises and practical assignments to apply theoretical concepts, including computer hardware installation, software setup, network configuration, and basic programming tasks. 9. Understanding of Professional Ethics and Communication: Identify ethical considerations in the IT field, including privacy, intellectual property, and responsible technology use. Exhibit effective communication skills for collaboration and documentation purposes.
Indicative Contents المحتويات الإرشادية	The indicative contents of the IT basics course for the computer department may include the following topics: 11. Introduction to Information Technology: o Definition and scope of information technology o Evolution and impact of IT on society and businesses o Ethical considerations and challenges in IT 12. Computer Systems and Hardware: o Components of a computer system (CPU, memory, storage, input/output devices) o Computer organization and architecture o Digital data representation (binary, hexadecimal) 13. Operating Systems: o Functions and types of operating systems (e.g., Windows, macOS, Linux) o Process management and multitasking o Memory management and virtual memory o File systems and file management 14. Software and Applications Development: o Programming languages and their characteristics (e.g., Java, Python, C++)

- Algorithms and problem-solving techniques
- Introduction to software development methodologies (e.g., waterfall, agile)

15. Networking Fundamentals:
    - Network topologies (e.g., bus, star, mesh)
    - Network protocols (e.g., TCP/IP, HTTP, DNS)
    - Local area networks (LANs) and wide area networks (WANs)
    - Network security and basic concepts of cybersecurity
16. Database Systems:
    - Introduction to database management systems (DBMS)
    - Relational database concepts and design principles
    - Structured Query Language (SQL) for data manipulation and retrieval
    - Basic database administration tasks and data integrity
17. Web Technologies:
    - Basics of web development (HTML, CSS, JavaScript)
    - Client-server architecture and web application deployment
    - Web design principles and usability considerations
    - Introduction to content management systems (CMS)
18. Information Security:
    - Fundamentals of information security and data protection
    - Common security threats and vulnerabilities
    - Authentication and access control mechanisms
    - Encryption and cryptographic techniques
19. Emerging Technologies:
    - Cloud computing and virtualization
    - Artificial intelligence and machine learning
    - Big data analytics and data science
    - Internet of Things (IoT) and its applications
20. Ethical and Legal Considerations:
    - Ethical issues in IT, including privacy and responsible use of technology

	○ Intellectual property rights and copyright laws ○ Cybersecurity laws and regulations ○ Data protection and privacy laws (e.g., GDPR, CCPA)

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	Learning and teaching strategies for the IT basics course for the Network department can include a combination of the following: 11. Active Learning: Incorporate activities that engage students actively in the learning process. This can include hands-on exercises, group projects, case studies, and discussions. Encourage students to apply the concepts they learn to real-world scenarios and problems. 12. Practical Exercises: Provide opportunities for students to practice and apply their knowledge. Assign programming exercises, database design projects, or networking simulations to reinforce understanding and develop practical skills. Offer guidance and feedback during the exercises to facilitate learning. 13. Real-World Examples: Use relevant and relatable examples from various industries and domains to illustrate the application of IT concepts. Demonstrate how technology is used in business processes, healthcare, finance, or other fields to make the concepts more tangible and meaningful. 14. Multimedia Resources: Supplement lectures and readings with multimedia resources such as videos, interactive tutorials, online demonstrations, and virtual labs. Visual and interactive content can help students grasp complex concepts and engage different learning styles. 15. Guest Speakers and Industry Connections: Invite guest speakers from the IT industry or academia to share their experiences, insights, and real-world applications of IT concepts. Establish connections with professionals working in the field to provide students with networking opportunities and industry perspectives. 16. Problem-Solving and Critical Thinking: Emphasize problem-solving and critical-thinking skills throughout the course. Encourage students to analyze and evaluate information, think creatively, and propose

solutions to IT-related challenges. Pose thought-provoking questions and scenarios to stimulate discussion and higher-level thinking.

17. Assessment Variety: Use a variety of assessment methods to evaluate students' understanding and mastery of the concepts. This can include quizzes, exams, projects, presentations, and portfolios. Incorporate both individual and group assessments to encourage collaboration and teamwork skills.
18. Current and Emerging Trends: Introduce students to current trends and emerging technologies in the field of information technology. Discuss their impact, potential applications, and challenges. Encourage students to explore and research these topics to stay updated with the evolving IT landscape.
19. Ethical and Legal Considerations: Incorporate discussions and assignments that focus on ethical and legal issues in IT. Promote discussions on responsible use of technology, privacy concerns, intellectual property rights, and cybersecurity ethics. Foster awareness of the ethical implications of IT in society.
20. Continuous Feedback and Support: Provide regular feedback on students' progress, both during in-class activities and through assignments. Offer opportunities for students to seek clarification and ask questions. Create a supportive learning environment that encourages open communication and collaboration.

These strategies promote active learning, practical application of knowledge, and engagement with the subject matter. They cater to different learning styles and encourage students to develop critical thinking, problem-solving, and communication skills necessary for success in the IT field.

Student Workload (SWL)

الحمل الدراسي للطالب محسوب لـ ١٥ اسبوعا

Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	63	**Structured SWL (h/w)** الحمل الدراسي المنتظم للطالب أسبوعيا	6
Unstructured SWL (h/sem)	62	**Unstructured SWL (h/w)**	6

الحمل الدراسي غير المنتظم للطالب خلال الفصل		الحمل الدراسي غير المنتظم للطالب أسبوعيا	
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	125		

Module Evaluation

تقييم المادة الدراسية

		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	**Quizzes**	2	10% (10)	5 and 10	LO #1, #2 and #10, #11
	Assignments	2	10% (10)	2 and 12	LO #3, #4 and #6, #7
	Projects / Lab.	1	10% (10)	Continuous	All
	Report	1	10% (10)	13	LO #5, #8 and #10
Summative assessment	**Midterm Exam**	2hr	10% (10)	7	LO #1 - #7
	Final Exam	3hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)

المنهاج الاسبوعي النظري

	Material Covered
Week 1	Introduction to the basic concepts of Information Technology (IT) and their applications.
Week 2	Understand the structure layers of the infrastructure model of Information technology environment, especially about end-users, operating system, computer network, and storage.
Week 3	Computer Hardware • Components of a computer system • Input and output devices

	• Computer peripherals and their functions
Week 4	Computer Software • Types of software: system software and application software • Operating systems and their features • Software installation and management
Week 4	Practical Skills - Operating Systems Basic concepts under IT infrastructure model layers • Installation and configuration of operating systems • File management operations • System administration tasks
Week 5	End-users concept underlying infrastructure model
Week 6	Networks and Connectivity-Basic concepts under IT infrastructure model layers • Network architectures: LAN, WAN, WLAN • Network protocols: TCP/IP, Ethernet, Wi-Fi
Week 7	Networks and Connectivity Basic concepts under IT infrastructure model layers • Network architectures: LAN, WAN, WLAN • Network protocols: TCP/IP, Ethernet, Wi-Fi
Week 8	Mid term Examination
Week 9	Storage concept underlying infrastructure model
Week 10	Introduction to Data Management Basic concepts under IT infrastructure model layers • Basics of databases and data management systems
Week 11	Introduction to Data Management Basic concepts under IT infrastructure model layers • Structured Query Language (SQL)
Week 12	Introduction to Web Development Basic concepts under IT infrastructure model layers • HTML and CSS fundamentals
Week 13	Introduction to Web Development Basic concepts under IT infrastructure layers • Web page creation and design principles

Week 14	Ethical Considerations Basic concepts under IT infrastructure layers • Ethics in the IT field: privacy, intellectual property, responsible technology use • Professional communication and documentation skills
Week 15	Week 15: Review • Review of key concepts covered throughout the course • Completion of final projects or assignments demonstrating understanding of IT basics
Week 16	**Preparatory week before the final Exam**

Delivery Plan (Weekly Lab. Syllabus) المنهاج الأسبوعي للمختبر	
	Material Covered
Week 1	Operating System Installation and Configuration o Objective: Enable students to install and configure an operating system. o Activities: Students will install an operating system of their choice (e.g., Windows, Linux) on a virtual machine or physical computer. They will configure settings, create user accounts, and explore basic system administration tasks.
Week 2	Introduction to use the terminal of Ubuntu Operating System
Week 3	Introduction to Ubuntu environment.
Week 4	Introduction to the Shell
Week 5	Navigation
Week 6	Navigation
Week 7	Exploring the System
Week 8	Mid term Exanimation
Week 9	Manipulating Files and Directories
Week 10	Permissions
Week 11	Processes

Week 12	Configuration and the Environment
Week 13	Storage Media
Week 14	Networking
Week 15	Networking

Learning and Teaching Resources

مصادر التعلم والتدريس

	Text	Available in the Library?
Required Texts		No
Recommended Texts		
Websites		

Grading Scheme

مخطط الدرجات

Group	Grade	التقدير	Marks %	Definition
Success Group (50 - 100)	**A -** Excellent	امتياز	90 - 100	Outstanding Performance
	B - Very Good	جيد جدا	80 - 89	Above average with some errors
	C - Good	جيد	70 - 79	Sound work with notable errors
	D - Satisfactory	متوسط	60 - 69	Fair but with major shortcomings
	E - Sufficient	مقبول	50 - 59	Work meets minimum criteria
Fail Group (0 – 49)	**FX –** Fail	راسب (قيد المعالجة)	(45-49)	More work required but credit awarded
	F – Fail	راسب	(0-44)	Considerable amount of work required

Note: Marks Decimal places above or below 0.5 will be rounded to the higher or lower full mark (for example a mark of 54.5 will be rounded to 55, whereas a mark of 54.4 will be rounded to 54. The University has a policy NOT to condone "near-pass fails" so the only adjustment to marks awarded by the original marker(s) will be the automatic rounding outlined above.

MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information

معلومات المادة الدراسية

Module Title	**Calculus**	**Module Delivery**	
Module Type	**Support**	☒ **Theory** ☒ **Lecture** ☒ **Lab** ☐ **Tutorial** ☐ **Practical** ☐ **Seminar**	
Module Code	**NT103**		
ECTS Credits	**5**		
SWL (hr/sem)	**125**		

Module Level	1	**Semester of Delivery**	1
Administering Department	NT	**College**	CSM
Module Leader	Name	**e-mail**	E-mail
Module Leader's Acad. Title	Professor	**Module Leader's Qualification**	Ph.D.
Module Tutor	Name (if available)	**e-mail**	E-mail
Peer Reviewer Name	Name	**e-mail**	E-mail
Scientific Committee Approval Date	01/06/2023	**Version Number**	1.0

Relation with other Modules

العلاقة مع المواد الدراسية الأخرى

Prerequisite module	None	**Semester**	
Co-requisites module	None	**Semester**	

Module Aims, Learning Outcomes and Indicative Contents

أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Objectives أهداف المادة الدراسية	• Understanding Limits: Students should develop a clear understanding of limits and their properties. They should be able to evaluate limits algebraically and graphically and comprehend the concept of continuity. • Calculating Derivatives: Students should learn to compute derivatives using basic rules, such as the power rule, product rule, quotient rule, and chain rule. They should understand the interpretation of derivatives as rates of change and be able to apply derivatives to solve problems involving optimization, related rates, and approximations. • Analyzing Functions: Students should be able to analyze and interpret functions using calculus tools. This includes determining intervals of increase and decrease, finding local extrema, identifying points of inflection, and sketching the graph of a function using differentiation. • Evaluating Integrals: Students should learn to evaluate definite and indefinite integrals. They should understand the concept of antiderivatives, basic integration rules, and techniques such as substitution and integration by parts. They should also be able to apply integrals to solve problems involving area, average value, and basic differential equations. • Understanding the Fundamental Theorem of Calculus: Students should grasp the concepts behind the Fundamental Theorem of Calculus and its implications. They should be able to use the Fundamental Theorem of Calculus to evaluate definite integrals and relate integrals to accumulation functions. • Developing Problem-Solving Skills: Students should enhance their problem-solving abilities by applying calculus concepts and techniques to a variety of real-world and mathematical problems. They should develop logical reasoning, critical thinking, and analytical skills to solve calculus problems effectively. • Enhancing Mathematical Communication: Students should develop the ability to communicate mathematical ideas and solutions clearly and effectively. They should be able to express their reasoning, use appropriate mathematical notation, and present their work in a well-organized manner.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	Upon successful completion of the Calculus course for the Networks department, students should be able to demonstrate the following learning outcomes: 1. Knowledge and Understanding: • Demonstrate a solid understanding of fundamental concepts in calculus, including limits, derivatives, and integrals. • Explain the relationship between functions, limits, and continuity. • Apply the concepts of calculus to solve problems related to rates of change, optimization, and approximation. 2. Computational Skills: • Perform algebraic manipulations and simplify expressions involving limits, derivatives, and integrals. • Compute limits of functions and evaluate derivatives using various techniques, such as the power rule, chain rule, and product rule. • Solve problems involving optimization, related rates, and approximations using calculus methods. 3. Application and Modeling: • Apply calculus concepts to real-world scenarios and model various physical and mathematical phenomena. • Interpret and analyze graphs, tables, and equations representing

	functions and their derivatives. • Use calculus to solve problems in areas such as physics, economics, engineering, and biology. 4. Critical Thinking and Problem-Solving: • Analyze and interpret problems to identify relevant mathematical concepts and apply appropriate calculus techniques. • Develop logical reasoning and problem-solving strategies to solve complex calculus problems. • Evaluate the reasonableness and accuracy of solutions and interpret their implications in practical contexts. 5. Mathematical Communication: • Express mathematical ideas and solutions clearly and accurately using appropriate mathematical language and notation. • Communicate mathematical reasoning and solution processes effectively through written explanations, diagrams, and graphs. • Present and communicate mathematical results in a coherent and organized manner. 6. Mathematical Reasoning and Proof: • Apply logical reasoning and mathematical proof techniques to justify mathematical statements and results in calculus. • Understand and construct mathematical arguments, including proofs of key calculus theorems and properties. • Recognize the importance of mathematical rigor and precision in calculus reasoning. 7. Technology and Calculus Tools: • Utilize technology, such as graphing calculators or computer software, to aid in visualizing and analyzing calculus concepts. • Apply appropriate technological tools to perform numerical computations, graph functions, and solve calculus problems. • Interpret and validate results obtained from technology tools in the context of calculus applications.
Indicative Contents المحتويات الإرشادية	The indicative contents of the Calculus course for the computer department may include the following topics: • Introduction to Calculus: • Basic concepts of functions, including domain, range, and graphing • Types of functions (polynomial, exponential, logarithmic, trigonometric) • The concept of a limit and its properties • Limits and Continuity: • Evaluating limits algebraically and graphically • Determining one-sided and infinite limits • Continuity of functions and its properties • Differentiation: • Definition of the derivative and its interpretation as a rate of change • Calculation of derivatives using basic rules (power rule, constant rule, sum and difference rules) • Derivatives of trigonometric, exponential, and logarithmic functions • Higher-order derivatives and their interpretation

	• Applications of Differentiation:
	  • Tangent lines and rates of change
	  • Optimization problems (finding maximum or minimum values)
	  • Related rates problems (finding rates of change of related quantities)
	  • Approximation using differentials and linearization
	• Techniques of Differentiation:
	  • Product rule and quotient rule
	  • Chain rule for composite functions
	  • Implicit differentiation
	  • Derivatives of inverse trigonometric functions
	• Curve Sketching:
	  • Analysis of functions, including intervals of increase/decrease, local extrema, and concavity
	  • Determining asymptotes, intercepts, and symmetry
	  • Sketching the graph of a function using differentiation and critical points
	• Integration:
	  • Antiderivatives and indefinite integrals
	  • Definite integrals and their interpretation as areas
	  • Basic integration rules (power rule, sum and difference rules)
	  • Techniques of integration (substitution, integration by parts)
	• Applications of Integration:
	  • Calculation of areas between curves
	  • Determining the average value of a function
	  • Finding the area of a region bounded by curves
	  • Solving basic differential equations
	• Fundamental Theorem of Calculus:
	  • Statement and application of the Fundamental Theorem of Calculus
	  • Evaluating definite integrals using the Fundamental Theorem of Calculus
	  • Area under a curve and accumulation functions
	• Numerical Methods:
	  • Approximating definite integrals using numerical methods (e.g., midpoint rule, trapezoidal rule)
	  • Simpson's rule for numerical integration
	  • Applications of numerical methods in practical contexts.

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	Learning and teaching strategies for the Wireless Sensor Networks course for the Network department can include a combination of the following: 1. Clear Explanation and Examples: Provide clear explanations of calculus concepts, definitions, and theorems. Use relatable examples and step-

by-step solutions to illustrate the application of concepts and problem-solving techniques.

2. Active Learning: Engage students in active learning experiences through in-class activities, group discussions, and problem-solving exercises. Encourage students to actively participate in the learning process by asking questions, working through problems, and explaining their reasoning.
3. Visual Representations: Utilize visual aids, such as graphs, diagrams, and animations, to illustrate calculus concepts. Visual representations can help students visualize functions, understand the geometric interpretation of calculus concepts, and grasp complex ideas more easily.
4. Real-world Applications: Relate calculus concepts to real-world applications to make them more meaningful and relevant to students. Show examples of how calculus is used in various fields, such as physics, economics, engineering, and biology, to solve practical problems and make predictions.
5. Practice and Feedback: Provide ample opportunities for students to practice solving calculus problems. Assign homework exercises, practice quizzes, and problem sets to reinforce understanding and develop problem-solving skills. Offer constructive feedback on student work to guide their learning and address common mistakes.
6. Technology Integration: Integrate technology tools, such as graphing calculators, mathematical software, or online interactive resources, to aid visualization, exploration, and computation in calculus. Demonstrate how technology can be used to enhance understanding and solve complex problems efficiently.
7. Conceptual Understanding: Emphasize the development of conceptual understanding alongside procedural skills. Encourage students to reason and explain concepts, connections, and theorems rather than relying solely on memorization and algorithms.
8. Collaborative Learning: Promote collaborative learning environments by incorporating group activities, peer discussions, and projects. Encourage students to work together, exchange ideas, and explain concepts to their peers. Collaborative learning can enhance problem-solving skills, critical thinking, and communication.
9. Formative Assessment: Use formative assessment strategies, such as quizzes, class discussions, and concept-check questions, to monitor student progress and identify areas of difficulty. Provide timely feedback to address misconceptions and guide further learning.
10. Office Hours and Support: Offer regular office hours and additional support sessions to provide students with opportunities for individual or small-group consultations. Address individual questions, clarify concepts, and provide personalized guidance to students who may require extra assistance.

Student Workload (SWL) الحمل الدراسي للطالب محسوب لـ ١٥ اسبوعا			
Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	48	**Structured SWL (h/w)** الحمل الدراسي المنتظم للطالب أسبوعيا	6
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	77	**Unstructured SWL (h/w)** الحمل الدراسي غير المنتظم للطالب أسبوعيا	6
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	125		

Module Evaluation
تقييم المادة الدراسية

		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	Quizzes	2	10% (10)	5 and 10	LO #1, #2 and #10, #11
	Assignments	2	10% (10)	2 and 12	LO #3, #4 and #6, #7
	Projects / Lab.	1	10% (10)	Continuous	All
	Report	1	10% (10)	13	LO #5, #8 and #10
Summative assessment	Midterm Exam	2hr	10% (10)	7	LO #1 - #7
	Final Exam	3hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)
المنهاج الاسبوعي النظري

	Material Covered
Week 1	• General Review: Real number and their properties, intervals, inequalities, Absolute value with its properties.
Week 2	• The Real Function and its graphs, domain and range.
Week 3	• Limits and continuity: definition, theorems, properties, types of limits.
Week 4	• Derivative of function: Theory of derivative, higher order derivative, Implicit derivative, Chain rule.
Week 5	• The integral: definite and indefinite integrals and applications.

Week 6	• Transcendental Functions: Exponential Function, Logarithmic Function with derivatives and integrals
Week 7	• Trigometric Function, Inverse Trigometric Function with derivatives and integrals
Week 8	• Mid – Term Examination
Week 9	• Matrices: definition, types and their operations
Week 10	• Determinants: definition, properties and applications, The Cofactor and the Inverse of matrix by Cofactor, Grammar's method
Week 11	• Laplace transformation: definition, examples
Week 12	• Inverse Laplace transformation: definition, examples
Week 13	• Series: definition, Taylor and Maclaurin series
Week 14	• Fourier series
Week 15	• Review the Course

Learning and Teaching Resources
مصادر التعلم والتدريس

	Text	Available in the Library?
Required Texts		No
Recommended Texts		
Websites		

Grading Scheme
مخطط الدرجات

Group	Grade	التقدير	Marks %	Definition
Success Group (50 - 100)	**A -** Excellent	امتياز	90 - 100	Outstanding Performance
	B - Very Good	جيد جدا	80 - 89	Above average with some errors
	C - Good	جيد	70 - 79	Sound work with notable errors
	D - Satisfactory	متوسط	60 - 69	Fair but with major shortcomings
	E - Sufficient	مقبول	50 - 59	Work meets minimum criteria
Fail Group (0 – 49)	**FX –** Fail	راسب (قيد المعالجة)	(45-49)	More work required but credit awarded
	F – Fail	راسب	(0-44)	Considerable amount of work required

Note: Marks Decimal places above or below 0.5 will be rounded to the higher or lower full mark (for example a mark of 54.5 will be rounded to 55, whereas a mark of 54.4 will be rounded to 54. The University has a policy NOT to condone "near-pass fails" so the only adjustment to marks awarded by the original marker(s) will be the automatic rounding outlined above.

MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information

معلومات المادة الدراسية

Module Title	**Logic Design Fundamentals**		**Module Delivery**	
Module Type	**Core**		☒ **Theory** ☒ **Lecture** ☒ **Lab** ☐ **Tutorial** ☐ **Practical** ☐ **Seminar**	
Module Code	**NT104**			
ECTS Credits	**7**			
SWL (hr/sem)	**175**			
Module Level	1	**Semester of Delivery**	1	
Administering Department	NT	**College**	CSM	
Module Leader	Name	**e-mail**	E-mail	
Module Leader's Acad. Title	Professor	**Module Leader's Qualification**	Ph.D.	
Module Tutor	Name (if available)	**e-mail**	E-mail	
Peer Reviewer Name	Name	**e-mail**	E-mail	
Scientific Committee Approval Date	18/06/2023	**Version Number**	1.0	

Relation with other Modules

العلاقة مع المواد الدراسية الأخرى

Prerequisite module	None	**Semester**	
Co-requisites module	None	**Semester**	

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Objectives أهداف المادة الدراسية	1. Understand the Basics of Digital Logic: Familiarize students with the fundamental concepts of digital logic, including binary number systems, logic gates, and Boolean algebra. 2. Learn Combinational Logic Design: Enable students to design and analyze combinational logic circuits using logic gates, multiplexers, decoders, and encoders. Develop skills in simplifying Boolean expressions and implementing logic functions. 3. Explore Sequential Logic Design: Introduce students to sequential logic circuits, including flip-flops, registers, and counters. Teach them to design and analyze sequential circuits using state diagrams and transition tables. 4. Develop Skills in Boolean Algebra Manipulation: Teach students the principles of Boolean algebra and logic simplification techniques, including Boolean laws, De Morgan's theorem, and Karnaugh maps. Enable them to simplify complex Boolean expressions. 5. Gain Proficiency in Circuit Analysis and Simulation: Provide students with the ability to analyze and simulate digital circuits using appropriate software tools. Help them understand the behavior of logic circuits and validate their designs. 6. Apply Design Methodologies: Introduce students to structured design methodologies for digital circuits, including the concept of hierarchical design, module reuse, and design documentation practices. 7. Foster Problem-Solving and Critical Thinking Skills: Encourage students to apply logical reasoning and critical thinking in solving complex problems related to digital logic design. Develop their ability to break down problems into smaller components and apply appropriate design techniques. 8. Enhance Practical Skills through Lab Exercises: Provide hands-on lab exercises where students can design, implement, and test digital logic circuits using hardware components and/or digital simulation software. Reinforce theoretical concepts through practical application. 9. Foster Collaboration and Communication Skills: Promote teamwork and effective communication skills through group projects and presentations. Encourage students to collaborate on circuit design and problem-solving activities.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	Upon completing the course, students should be able to: 1. Understand the fundamental concepts of digital logic design, including binary number systems, logic gates, and Boolean algebra. 2. Comprehend the principles and characteristics of combinational and sequential logic circuits. 3. Explain the behavior and operation of various digital components, such as flip-flops, registers, and counters. 4. Understand the different types of memory devices and programmable logic devices. 5. Design and implement combinational logic circuits using logic gates, multiplexers, decoders, and encoders. 6. Simplify Boolean expressions and optimize logic functions using Boolean algebra and logic simplification techniques.

	7. Design and analyze sequential logic circuits using state diagrams, transition tables, and timing diagrams. 8. Implement digital circuits using programmable logic devices (PLDs) and understand their programming and configuration. 9. Apply logical reasoning and critical thinking skills to solve problems related to digital logic design. 10. Design, implement, and test digital logic circuits using hardware components and/or digital simulation software. 11. Use appropriate software tools for circuit simulation, validation, and analysis. 12. Work effectively in teams to collaboratively design and implement digital logic circuits. 13. Collaborate and contribute to group projects and discussions related to digital logic design.
Indicative Contents المحتويات الإرشادية	The following are indicative contents that may be covered in the course: 1. Introduction to Digital Logic: - Number systems: binary, decimal, octal, and hexadecimal - Boolean algebra: logic operators, truth tables, and laws - Logic gates: AND, OR, NOT, XOR, NAND, NOR, and XNOR gates 2. Combinational Logic Design: - Combinational circuits: design and analysis - Boolean functions: expressions, canonical forms, and simplification techniques - Karnaugh maps: truth table to K-map conversion and simplification - Implementation of combinational circuits using logic gates 3. Combinational Circuits: - Multiplexers: operation, design, and applications - Demultiplexers: operation, design, and applications - Encoders: operation, design, and applications - Decoders: operation, design, and applications 4. Sequential Logic Design: - Flip-flops: SR, D, JK, and T flip-flops

Learning and Teaching Strategies
استراتيجيات التعلم والتعليم

Strategies	Learning and teaching strategies for the Network department can include a combination of the following: 21. Lectures: Engage students through informative lectures that cover theoretical concepts and provide an overview of key topics. Use multimedia resources, visuals, and real-world examples to enhance understanding. 22. Hands-on Labs: Provide practical lab sessions where students can apply their knowledge and skills acquired in lectures. These labs can involve hardware assembly, software installation, network configuration, programming exercises, and troubleshooting. 23. Group Discussions and Collaborative Learning: Encourage group discussions and collaborative activities to foster interaction and knowledge sharing among students. Assign group projects or case studies that require teamwork and problem-solving. 24. Guest Speakers and Industry Experts: Invite guest speakers from the industry to share their experiences, insights, and the latest trends in the IT field. This can provide students with a real-world perspective and inspire them to explore various career paths. 25. Online Resources and Multimedia: Utilize online resources, interactive tutorials, and multimedia materials to supplement learning. This can include video lectures, online quizzes, virtual labs, and interactive modules. 26. Assignments and Projects: Assign individual and group projects that require students to apply their knowledge and skills to solve real-world problems or complete practical tasks. This promotes critical thinking, problem-solving, and practical application of concepts. 27. Assessments and Feedback: Conduct regular assessments, quizzes, and examinations to evaluate students' understanding of the course material. Provide timely and constructive feedback to help students identify areas of improvement. 28. Online Discussion Forums and Communication Platforms: Establish online discussion forums or communication platforms where students can ask questions, share resources, and engage in discussions outside of the classroom.

Student Workload (SWL) الحمل الدراسي للطالب محسوب لـ ١٥ اسبوعا			
Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	108	**Structured SWL (h/w)** الحمل الدراسي المنتظم للطالب أسبوعيا	7
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	92	**Unstructured SWL (h/w)** الحمل الدراسي غير المنتظم للطالب أسبوعيا	6
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	**200**		

Module Evaluation

تقييم المادة الدراسية

		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	Quizzes	2	10% (10)	5 and 10	LO #1, #2 and #10, #11
	Assignments	2	10% (10)	2 and 12	LO #3, #4 and #6, #7
	Projects / Lab.	1	10% (10)	Continuous	All
	Report	1	10% (10)	13	LO #5, #8 and #10
Summative assessment	Midterm Exam	2hr	10% (10)	7	LO #1 - #7
	Final Exam	3hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)

المنهاج الاسبوعي النظري

	Material Covered
Week 1	Introduction to Digital Logic Design - Digital logic levels and signals
Week 2-3	Introduction to Number systems - Binary - Decimal - Octal - Hexadecimal
Week 4-5	Introduction to logic gates and truth table (AND, OR, NOT, NAND, NOT, EX-OR, and EX-NOR)
Week 6	Boolean Algebra - Boolean variables and expressions - Boolean laws and theorems
Week 7	Simplification of Boolean expressions
Week 8-9	Combinational Logic Gate Circuits and truth tables
Week 10	Designing and analyzing combinational circuits
Week 11	Multiplexers and De-multiplexers
Week 12	Karnaugh maps and simplification techniques
Week 13	Arithmetic Circuits - Binary addition and subtraction circuits
Week 14	Binary-coded decimal (BCD) and binary-to-BCD conversion
Week 15	Flip-Flops

Delivery Plan (Weekly Lab. Syllabus) المنهاج الاسبوعي للمختبر	
	Material Covered
Week 1	Introduction to Logic Gates (AND, OR,NOT, NAND, NOR, EX-OR, and EX-NOR)
Week 2	Construct and verify the truth tables for basic logic gates (AND, OR, NOT).
Week 3	Build logic gate circuits using breadboards and test their functionality.
Week 4	Boolean Algebra and Logic Simplification - Simplify Boolean expressions using Boolean algebra laws and theorems.
Week 5	Implement simplified expressions using logic gates and verify the results.
Week 6-7	Combinational Logic Circuits - Design and implement a half-adder circuit using logic gates. - Build a full-adder circuit and test its functionality.
Week 8	- Design and construct a 4-bit binary adder-subtractor circuit. - Build a BCD adder circuit and verify its functionality.
Week 9-10	Combinational Logic Design - Design and build a 4-bit binary-to-BCD converter using combinational logic.
Week 11	- Construct and verify the functionality of a 4-bit magnitude comparator.
Week 12-13	Multiplexers and Decoders 1. Build a 4-to-1 multiplexer and test its operation using different input combinations. 2. Design and construct a 3-to-8 decoder using basic logic gates.
Week 14-15	Sequential Logic Circuits - Construct and verify the functionality of a D flip-flop using basic components.

Learning and Teaching Resources مصادر التعلم والتدريس		
	Text	**Available in the Library?**
Required Texts	Select a comprehensive logic design textbook that covers the fundamental concepts, principles, and techniques of digital logic design. Examples include "Digital Design" by M. Morris Mano and Michael D. Ciletti or "Digital Logic and Computer Design" by M. Morris Mano.	**Yes**
Recommended Texts	"Digital Design" by M. Morris Mano and Michael D. Ciletti **or** "Digital Logic and Computer Design" by M. Morris Mano	
Websites		

Grading Scheme مخطط الدرجات				
Group	**Grade**	التقدير	**Marks %**	**Definition**
Success Group (50 - 100)	**A** - Excellent	امتياز	90 - 100	Outstanding Performance
	B - Very Good	جيد جدا	80 - 89	Above average with some errors
	C - Good	جيد	70 - 79	Sound work with notable errors
	D - Satisfactory	متوسط	60 - 69	Fair but with major shortcomings
	E - Sufficient	مقبول	50 - 59	Work meets minimum criteria
Fail Group (0 – 49)	**FX** – Fail	راسب (قيد المعالجة)	(45-49)	More work required but credit awarded
	F – Fail	راسب	(0-44)	Considerable amount of work required

Note: Marks Decimal places above or below 0.5 will be rounded to the higher or lower full mark (for example a mark of 54.5 will be rounded to 55, whereas a mark of 54.4 will be rounded to 54. The University has a policy NOT to condone "near-pass fails" so the only adjustment to marks awarded by the original marker(s) will be the automatic rounding outlined above.

MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information

معلومات المادة الدراسية

Module Title	**English 1**	**Module Delivery**	
Module Type	**Support**	☒ **Theory** ☒ **Lecture** ☐ **Lab** ☐ **Tutorial** ☐ **Practical** ☐ **Seminar**	
Module Code	**UOM102**		
ECTS Credits	**2**		
SWL (hr/sem)	**50**		

Module Level	1	**Semester of Delivery**	1
Administering Department	NT	**College**	CSM
Module Leader	Name	**e-mail**	E-mail
Module Leader's Acad. Title		**Module Leader's Qualification**	
Module Tutor	Name (if available)	**e-mail**	E-mail
Peer Reviewer Name	Name	**e-mail**	E-mail
Scientific Committee Approval Date		**Version Number**	1.0

Relation with other Modules

العلاقة مع المواد الدراسية الأخرى

Prerequisite module	None	**Semester**	
Co-requisites module	None	**Semester**	

Module Aims, Learning Outcomes and Indicative Contents

أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية

Module Objectives أهداف المادة الدراسية	1. Language Proficiency: Develop basic language proficiency in English, including listening, speaking, reading, and writing skills. 2. Grammar: Understand and apply basic grammatical structures, including parts of speech, sentence formation, verb tenses, subject-verb agreement, and basic sentence patterns. 3. Vocabulary Building: Expand vocabulary through learning and practicing common words, synonyms, antonyms, idioms, phrasal verbs, and collocations. 4. Reading Comprehension: Improve reading skills by understanding main ideas, supporting details, making inferences, and analyzing texts of

	varying complexity. 5. Listening Comprehension: Enhance listening skills by understanding spoken English, including conversations, lectures, and presentations, and extracting key information. 6. Speaking Skills: Develop oral communication skills through practicing pronunciation, participating in conversations, giving presentations, and expressing opinions. 7. Writing Skills: Enhance writing abilities by practicing sentence construction, paragraph development, descriptive writing, narrative writing, and basic essay structure. 8. Cultural Awareness: Gain cultural understanding and appreciation through exposure to English-language literature, media, and diverse perspectives. 9. Study Skills: Develop effective study strategies, note-taking techniques, and time management skills for English language learning. 10. Assessment: Demonstrate language proficiency through quizzes, tests, presentations, writing assignments, and class participation.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	Upon successful completion of the English 1 course for the Networks department, students should be able to demonstrate the following learning outcomes: 1. Demonstrate basic proficiency in listening, speaking, reading, and writing skills in English. 2. Apply grammatical structures accurately to communicate effectively in written and spoken English. 3. Expand their vocabulary and use appropriate words and phrases in various contexts. 4. Comprehend and analyze written texts of different genres, including articles, short stories, and essays. 5. Understand spoken English in various situations, such as conversations, lectures, and presentations. 6. Engage in effective verbal communication, express opinions, and participate in discussions. 7. Write clear and coherent sentences, paragraphs, and short essays using proper organization and language conventions. 8. Develop cultural awareness and sensitivity to different cultural perspectives reflected in English literature and media. 9. Apply effective study skills, including note-taking, time management, and self-assessment techniques. 10. Demonstrate language proficiency through assessments, including quizzes, exams, presentations, and writing assignments.
Indicative Contents المحتويات الإرشادية	The indicative contents for the English 1 course may include the following topics: 1. Introduction to English Language: o Basic grammar rules and sentence structure o Parts of speech: nouns, verbs, adjectives, adverbs, etc. o Simple sentence construction and punctuation 2. Vocabulary Building:

	○ Commonly used words and expressions ○ Word formation: prefixes, suffixes, and root words ○ Synonyms, antonyms, and idiomatic expressions 3. Reading Comprehension: ○ Developing reading skills through texts of varying difficulty ○ Understanding main ideas, supporting details, and inference ○ Practicing skimming and scanning techniques 4. Writing Skills: ○ Paragraph writing: topic sentences, supporting details, and concluding sentences ○ Sentence structure and paragraph coherence ○ Developing basic writing skills: descriptive, narrative, and expository writing 5. Listening Skills: ○ Listening to and understanding spoken English in different contexts ○ Note-taking and summarizing information from spoken sources ○ Developing listening comprehension through audio materials and dialogues 6. Speaking Skills: ○ Basic conversation skills: greetings, introductions, and simple dialogues ○ Pronunciation and intonation practice ○ Participating in group discussions and oral presentations 7. Cultural Awareness: ○ Exploring English-speaking countries and their cultures ○ Understanding cultural differences and norms in communication 8. Language Practice and Activities: ○ Role plays, pair work, and group activities to practice language skills ○ Language games, quizzes, and interactive exercises for reinforcement These indicative contents provide a general overview of the topics and skills covered in the English 1 course, focusing on developing foundational language skills in reading, writing, listening, and speaking.

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	The learning and teaching strategies for the English 1 course aim to create an engaging and interactive learning environment where students can actively participate and develop their language skills. Some effective strategies include: 1. Communicative Approach: Emphasizing the use of English for meaningful communication, allowing students to practice and apply language skills in real-life situations through role plays, pair work, and group activities.

	2. Task-based Learning: Providing students with practical tasks and projects that require them to use English to achieve specific goals, fostering critical thinking, problem-solving, and collaboration skills. 3. Multi-modal Learning: Integrating various learning resources such as textbooks, audio recordings, videos, and online materials to cater to different learning styles and enhance comprehension and language acquisition. 4. Scaffolded Instruction: Breaking down complex language concepts into manageable steps, providing clear instructions, and gradually increasing the level of difficulty to ensure students' understanding and progress. 5. Formative Assessment: Implementing regular quizzes, assignments, and in-class activities to gauge students' understanding and provide timely feedback for improvement. 6. Technology Integration: Utilizing digital tools and resources, such as language learning apps, online dictionaries, and multimedia platforms, to enhance language practice, vocabulary acquisition, and listening comprehension. 7. Authentic Materials: Exposing students to authentic English materials, such as news articles, short stories, and videos, to develop their reading and listening skills and expose them to real-world language use. 8. Error Correction and Feedback: Providing constructive feedback and error correction to guide students in improving their language accuracy and fluency, both in written and spoken English. 9. Cultural Immersion: Incorporating cultural activities, discussions, and projects to promote intercultural understanding and awareness of different English-speaking cultures. By employing these strategies, the English 1 course aims to create an engaging and effective learning environment that fosters students' language proficiency, confidence, and communication skills in English.

Student Workload (SWL) الحمل الدراسي للطالب محسوب لـ ١٥ اسبوعا			
Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	32	**Structured SWL (h/w)** الحمل الدراسي المنتظم للطالب أسبوعيا	4
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	43	**Unstructured SWL (h/w)** الحمل الدراسي غير المنتظم للطالب أسبوعيا	4
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	**75**		

Module Evaluation

تقييم المادة الدراسية

		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	Quizzes	2	10% (10)	5 and 10	LO #1, #2 and #10, #11
	Assignments	2	10% (10)	2 and 12	LO #3, #4 and #6, #7
	Projects / Lab.	1	10% (10)	Continuous	All
	Report	1	10% (10)	13	LO #5, #8 and #10
Summative assessment	Midterm Exam	2hr	10% (10)	7	LO #1 - #7
	Final Exam	2hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)

المنهاج الاسبوعي النظري

	Material Covered
Week 1	Week 1: Introduction to English 1, course overview, and language assessment.
Week 2	Week 2: Grammar: Parts of speech, sentence structure, and basic sentence patterns.
Week 3	Week 3: Vocabulary Building: Basic word formation, synonyms, antonyms, and context clues.
Week 4	Week 4: Reading Comprehension: Developing reading strategies, understanding main ideas, and supporting details.
Week 5	Week 5: Listening Comprehension: Listening for information, note-taking, and understanding spoken dialogues.
Week 6	Week 6: Speaking Skills: Introducing oneself, asking and answering questions, and participating in simple conversations.
Week 7	Week 7: Writing Skills: Sentence construction, paragraph development, and descriptive writing.
Week 8	Week 8: Grammar: Verb tenses, subject-verb agreement, and verb forms.
Week 9	Week 9: Vocabulary Expansion: Idioms, phrasal verbs, and collocations.
Week 10	Week 10: Reading Comprehension: Inferring meaning, making predictions, and analyzing texts.
Week 11	Week 11: Listening Comprehension: Identifying main ideas, understanding specific details, and listening for inference.
Week 12	Week 12: Speaking Skills: Giving opinions, expressing agreement/disagreement, and presenting short talks.
Week 13	Week 13: Writing Skills: Narrative writing, writing emails, and basic essay structure.
Week 14	Week 14: Grammar: Modals, conditionals, and reported speech.
Week 15	Week 15: Review and Assessment: Recap of course topics, practice exercises, and final assessment.

Week 16	**Preparatory week before the final Exam**

Delivery Plan (Weekly Lab. Syllabus) المنهاج الاسبوعي للمختبر	
	Material Covered
Week 1	○
Week 2	○
Week 3	○
Week 4	○
Week 5	○
Week 6	○
Week 7	○

Learning and Teaching Resources مصادر التعلم والتدريس		
	Text	**Available in the Library?**
Required Texts		
Recommended Texts		
Websites		

Grading Scheme

مخطط الدرجات

Group	**Grade**	التقدير	**Marks %**	**Definition**
Success Group (50 - 100)	**A -** Excellent	امتياز	90 - 100	Outstanding Performance
	B - Very Good	جيد جدا	80 – 89	Above average with some errors
	C - Good	جيد	70 – 79	Sound work with notable errors
	D - Satisfactory	متوسط	60 – 69	Fair but with major shortcomings
	E - Sufficient	مقبول	50 – 59	Work meets minimum criteria
Fail Group (0 – 49)	**FX –** Fail	راسب (قيد المعالجة)	(45-49)	More work required but credit awarded
	F – Fail	راسب	(0-44)	Considerable amount of work required

Note: Marks Decimal places above or below 0.5 will be rounded to the higher or lower full mark (for example a mark of 54.5 will be rounded to 55, whereas a mark of 54.4 will be rounded to 54. The University has a policy NOT to condone "near-pass fails" so the only adjustment to marks awarded by the original marker(s) will be the automatic rounding outlined above.

MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information

معلومات المادة الدراسية

Module Title	**Democracy and Human Rights**	**Module Delivery**	
Module Type	**S**	☒ **Theory** ☒ **Lecture** ☐ **Lab** ☐ **Tutorial** ☐ **Practical** ☐ **Seminar**	
Module Code	**UOM104**		
ECTS Credits	**2**		
SWL (hr/sem)	**50**		
Module Level	1	**Semester of Delivery**	1
Administering Department	NT	**College**	CSM
Module Leader		**e-mail**	
Module Leader's Acad. Title		**Module Leader's Qualification**	
Module Tutor	Name (if available)	**e-mail**	E-mail
Peer Reviewer Name	Name	**e-mail**	E-mail
Scientific Committee Approval Date		**Version Number**	1.0

Relation with other Modules

العلاقة مع المواد الدراسية الأخرى

Prerequisite module	None	**Semester**	
Co-requisites module	None	**Semester**	

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Objectives أهداف المادة الدراسية	The course aims to introduce human rights in order to defend human dignity and contribute to changing human life for the better regarding: change in values and feelings - and change in behavior, as well as promoting the idea of social justice and strengthening the link between the individual and the group and the state and its institutions, and developing monitoring skills Violations, dealing with violators, supporting the skills of understanding human rights issues, in addition to enhancing ways to participate in public affairs - citizenship.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	1. Human rights are a set of fundamental entitlements and freedoms that are inherent to all individuals, regardless of their nationality, race, gender, religion, or any other characteristic. 2. They are based on the principles of dignity, equality, and respect for the inherent worth and value of every human being.
Indicative Contents المحتويات الإرشادية	Human rights are universal, meaning they apply to everyone, everywhere, without discrimination. They encompass civil, political, economic, social, and cultural rights, and are often codified in international and national legal frameworks. Civil and political rights include the right to life, liberty, and security of person; freedom of expression, assembly, and association; the right to a fair trial; and protection against torture, arbitrary arrest, and discrimination.

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	Civil and political rights include the right to life, liberty, and security of person; freedom of expression, assembly, and association; the right to a fair trial; and protection against torture, arbitrary arrest, and discrimination.

Student Workload (SWL) الحمل الدراسي للطالب محسوب لـ ١٥ اسبوعا			
Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	32	**Structured SWL (h/w)** الحمل الدراسي المنتظم للطالب أسبوعيا	1
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	18	**Unstructured SWL (h/w)** الحمل الدراسي غير المنتظم للطالب أسبوعيا	1
Total SWL (h/sem)	**50**		

الحمل الدراسي الكلي للطالب خلال الفصل	

Module Evaluation
تقييم المادة الدراسية

		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	Quizzes	2	10% (10)	5 and 10	LO #1, #2 and #10, #11
	Assignments	2	10% (10)	2 and 12	LO #3, #4 and #6, #7
	Projects / Lab.	1	10% (10)	Continuous	All
	Report	1	10% (10)	13	LO #5, #8 and #10
Summative assessment	Midterm Exam	2hr	10% (10)	7	LO #1 - #7
	Final Exam	3hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)
المنهاج الاسبوعي النظري

	Material Covered
Week 1	جذور حقوق الانسان وتطورها في التاريخ البشري
Week 2	حقوق الانسان: التحديد والتعريف والضمانات
Week 3	محتوى: الحريات العامة
Week 4	النظرية العامة للحريات العامة
Week 5	النظام القانوني للحريات العامة
Week 6	ضمانات الحرية العامة
Week 7	ضمانات الحرية العامة
Week 8	مفهوم المساواة
Week 9	مبادئ الحريات العامة تفصيليا
Week 10	حرية الأمن والشعور والاطمئنان
Week 11	حريات الفكرية
Week 12	قانون الفصل بين الدولة والكنيسة
Week 13	حرية العمل
Week 14	حرية التجارة والصناعة
Week 15	حرية التجارة والصناعة المبحث الأول: الاحزاب السياسية والحريات العامة

	المبحث الثاني: الحريات العامة في العالم الثالث المبحث الثالث:التقدم العلمي والتقني والحريات العامة

Learning and Teaching Resources مصادر التعلم والتدريس		
	Text	**Available in the Library?**
Required Texts	د.امير عبد العزيز، حقوق الأنسان في الأسلام	NO
Recommended Texts	نسرين مجد عبده حسونة،2015 ، حقوق الأنسان... المفهوم والخصائص والتصنيفات والمصادر	NO
Websites		

Grading Scheme مخطط الدرجات				
Group	**Grade**	التقدير	**Marks %**	**Definition**
Success Group (50 - 100)	**A** - Excellent	امتياز	90 - 100	Outstanding Performance
	B - Very Good	جيد جدا	80 - 89	Above average with some errors
	C - Good	جيد	70 - 79	Sound work with notable errors
	D - Satisfactory	متوسط	60 - 69	Fair but with major shortcomings
	E - Sufficient	مقبول	50 - 59	Work meets minimum criteria
Fail Group (0 – 49)	**FX** – Fail	راسب (قيد المعالجة)	(45-49)	More work required but credit awarded
	F – Fail	راسب	(0-44)	Considerable amount of work required

Note: Marks Decimal places above or below 0.5 will be rounded to the higher or lower full mark (for example a mark of 54.5 will be rounded to 55, whereas a mark of 54.4 will be rounded to 54. The University has a policy NOT to condone "near-pass fails" so the only adjustment to marks awarded by the original marker(s) will be the automatic rounding outlined above.

MODULE DESCRIPTION FORM
نموذج وصف المادة الدراسية

Module Information معلومات المادة الدراسية		
Module Title	**Problems Solving & Programming II**	**Module Delivery**
Module Type	**Core**	☒ **Theory**

Module Code	**NT107**	☒ **Lecture** ☒ **Lab** ☐ **Tutorial** ☐ **Practical** ☐ **Seminar**
ECTS Credits	**6**	
SWL (hr/sem)	**150**	

Module Level	1	**Semester of Delivery**	2
Administering Department	NT	**College**	CSM
Module Leader	Name	**e-mail**	E-mail
Module Leader's Acad. Title	r	**Module Leader's Qualification**	Ph.D.
Module Tutor	Name (if available)	**e-mail**	E-mail
Peer Reviewer Name	Name	**e-mail**	E-mail
Scientific Committee Approval Date		**Version Number**	1.0

Relation with other Modules

العلاقة مع المواد الدراسية الأخرى

Prerequisite module	NT102	**Semester**	1
Co-requisites module	None	**Semester**	

Module Aims, Learning Outcomes and Indicative Contents

أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية

Module Objectives
أهداف المادة الدراسية

1. Understanding Object-Oriented Programming (OOP) Concepts:
    1. Define and explain the fundamental concepts of OOP, such as classes, objects, encapsulation, inheritance, and polymorphism.
    2. Understand the benefits and advantages of using OOP in software development.
2. Mastering C++ Syntax and Language Features:
    1. Acquire a solid understanding of C++ syntax, data types, control structures, functions, and basic input/output operations.
    2. Familiarize students with C++ libraries, particularly the Standard Template Library (STL).
3. Implementing Classes and Objects in C++:

	1. Learn how to define classes, create objects, and understand the relationship between classes and objects. 2. Understand the concepts of constructors, destructors, and member functions. 4. Encapsulation and Data Hiding: 1. Understand the principles of encapsulation and data hiding to create robust and maintainable code. 2. Implement access specifiers (public, private, protected) to control the visibility of class members. 5. Inheritance and Polymorphism: 1. Understand the concept of inheritance and its role in code reuse and extensibility. 2. Implement inheritance relationships and demonstrate knowledge of base and derived classes. 3. Learn about polymorphism and the use of virtual functions to achieve runtime polymorphism. 6. Advanced C++ Features: 1. Explore advanced features of C++, such as templates, operator overloading, and exception handling. 2. Understand smart pointers and memory management techniques. 7. Object-Oriented Design Principles: 1. Learn and apply key design principles, such as SOLID (Single Responsibility, Open-Closed, Liskov Substitution, Interface Segregation, Dependency Inversion) and DRY (Don't Repeat Yourself). 2. Understand the importance of designing reusable, modular, and maintainable code. 8. Practical Application and Problem-Solving: 1. Apply object-oriented concepts and C++ programming skills to solve real-world problems. 2. Design and implement larger projects using object-oriented design principles. 9. Debugging and Testing: 1. Develop skills in debugging C++ code and troubleshooting common errors. 2. Learn and apply effective testing techniques to ensure code correctness. 10. Collaboration and Software Development Best Practices: 1. Develop skills in collaborative software development, including version control systems (e.g., Git) and team-based coding practices. 2. Understand the importance of writing clean, readable, and well-documented code.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	Upon completing the course, students should be able to: 1. Knowledge and Understanding: 1. Demonstrate a solid understanding of the fundamental concepts of object-oriented programming, including classes, objects, inheritance, polymorphism, encapsulation, and data hiding. 2. Explain the benefits and advantages of using object-oriented programming in software development.

	3. Understand the syntax, language features, and libraries of C++, particularly the Standard Template Library (STL). 2. Programming Skills: 1. Develop proficiency in writing C++ code using appropriate syntax, data types, control structures, functions, and input/output operations. 2. Implement classes and objects in C++ and apply object-oriented design principles to create robust and maintainable code. 3. Utilize inheritance and polymorphism to achieve code reuse, extensibility, and runtime flexibility. 4. Demonstrate proficiency in advanced C++ features, such as templates, operator overloading, and exception handling. 5. Apply debugging techniques to identify and resolve errors in C++ code. 3. Problem-Solving and Application: 1. Analyze real-world problems and design appropriate solutions using object-oriented programming principles. 2. Apply object-oriented design techniques to model and solve complex software problems. 3. Develop and implement larger projects using object-oriented programming concepts and practices. 4. Use appropriate testing techniques to ensure code correctness and reliability. 4. Design and Documentation: 1. Design software solutions using object-oriented analysis and design techniques. 2. Apply software engineering principles to create well-designed, modular, and reusable code. 3. Write clear and well-documented code that follows established coding standards. 4. Document and communicate the design and functionality of software solutions effectively. 5. Collaboration and Professionalism: 1. Work effectively in teams to develop software solutions, utilizing version control systems and collaborative coding practices. 2. Demonstrate professionalism, ethical behavior, and effective communication skills in a software development context. 3. Apply software development best practices, including code reviews, testing, and project management techniques.
Indicative Contents المحتويات الإرشادية	21. Introduction to Object-Oriented Programming (OOP): o Overview of procedural programming vs. object-oriented programming. o Key concepts of OOP: classes, objects, encapsulation, inheritance, and polymorphism. o Benefits and advantages of using OOP in software development. 22. Introduction to C++: o Basics of C++ programming language: syntax, data types, control structures, functions.

	○ Input/output operations using streams. ○ Pointers and memory management. 23. Classes and Objects in C++: ○ Defining classes and objects. ○ Member variables and member functions. ○ Constructors and destructors. ○ Access specifiers: public, private, and protected. ○ Encapsulation and data hiding. 24. Inheritance and Polymorphism: ○ Inheritance hierarchy and base/derived classes. ○ Overriding member functions. ○ Virtual functions and dynamic polymorphism. ○ Abstract classes and pure virtual functions. ○ Polymorphism with pointers and references. 25. Advanced Topics in C++: ○ Templates and generic programming. ○ Operator overloading. ○ Exception handling. ○ Smart pointers and memory management. 26. Object-Oriented Design Principles: ○ SOLID principles (Single Responsibility, Open-Closed, Liskov Substitution, Interface Segregation, Dependency Inversion). ○ Design patterns: factory, observer, strategy, etc. ○ Designing and implementing reusable and modular code. 27. Standard Template Library (STL): ○ Overview of STL containers: vectors, lists, maps, etc. ○ STL algorithms: sorting, searching, etc. ○ Iterators and generic algorithms. 28. File Handling and Input/Output Streams: ○ Reading from and writing to files. ○ Input/output streams and stream manipulators. 29. Software Development Practices: ○ Debugging techniques and tools. ○ Testing and test-driven development. ○ Version control systems (e.g., Git) and collaborative coding practices. 30. Practical Application and Projects: ○ Implementing small to medium-sized projects using object-oriented principles and C++. ○ Applying object-oriented design techniques to solve real-world problems. ○ Project management and documentation.

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	Learning and teaching strategies for the IT basics course for the Network

	department can include a combination of the following: 29. Hands-on Programming Assignments: Provide students with regular programming assignments that require them to apply object-oriented concepts in C++. These assignments should progressively increase in complexity to reinforce their understanding and skills. 30. Project-Based Learning: Assign larger projects that allow students to design and implement software solutions using object-oriented principles. These projects should simulate real-world scenarios and require students to apply their knowledge in a practical context. 31. Code Reviews and Peer Feedback: Encourage students to review and provide feedback on each other's code. This promotes collaboration, exposes students to different coding styles, and helps them learn from each other's approaches. 32. Interactive Coding Sessions: Conduct interactive coding sessions where students can participate in live coding exercises or solve programming problems together as a class. This allows for immediate feedback and discussion, fostering active learning. 33. Use of Visualizations and Diagrams: Utilize visual aids such as diagrams, UML (Unified Modeling Language) representations, and flowcharts to illustrate object-oriented concepts and relationships. Visualizations help students grasp abstract concepts and improve their understanding of class hierarchies and interactions. 34. Code Walkthroughs and Examples: Walkthrough well-commented code examples to demonstrate the implementation of various object-oriented concepts in C++. This helps students understand how to apply these concepts in practice and promotes good coding practices. 35. Classroom Discussions and Debates: Engage students in discussions and debates on topics related to object-oriented programming. Encourage critical thinking and analysis of different design choices, design patterns, and trade-offs in software development. 36. Guest Lectures and Industry Insights: Invite industry professionals or experts to deliver guest lectures, sharing their experiences and insights on object-oriented programming in the real world. This exposes students to practical applications of the concepts they are learning and provides valuable industry perspectives. 37. Online Resources and Tutorials: Share supplementary online resources, tutorials, and interactive coding platforms that allow students to practice and reinforce their learning outside of the classroom. Online resources can include coding challenges, tutorials, and videos that provide additional explanations and examples. 38. Assessments and Feedback: Provide regular assessments, such as quizzes and exams, to gauge students' understanding of object-oriented programming concepts in C++. Provide constructive feedback on their work to guide their learning and improvement.

Student Workload (SWL)

الحمل الدراسي للطالب محسوب لـ ١٥ اسبوعا			
Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	78	**Structured SWL (h/w)** الحمل الدراسي المنتظم للطالب أسبوعيا	6
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	72	**Unstructured SWL (h/w)** الحمل الدراسي غير المنتظم للطالب أسبوعيا	8
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	**150**		

Module Evaluation
تقييم المادة الدراسية

		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	**Quizzes**	2	10% (10)	5 and 10	LO #1, #2 and #10, #11
	Assignments	2	10% (10)	2 and 12	LO #3, #4 and #6, #7
	Projects / Lab.	1	10% (10)	Continuous	All
	Report	1	10% (10)	13	LO #5, #8 and #10
Summative assessment	**Midterm Exam**	2hr	10% (10)	7	LO #1 - #7
	Final Exam	3hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)
المنهاج الاسبوعي النظري

	Material Covered
Week 1	Review to the Introduction to Problem Solving and Programming I
Week 2-3	Structure Compound Data types
Week 4-5	String Manipulation
Week 6	Dynamic Memory Allocation
Week 7-8	Files
Week 9	Mid Term Examination

Week 10	Exception Handling
Week 11-14	Prepare Mini Project
Week 15	Revision and Review

Delivery Plan (Weekly Lab. Syllabus) المنهاج الاسبوعي للمختبر	
	Material Covered
Week 1	Week 1: Review of Introduction to Problem Solving and Programming I • Recap of problem-solving techniques and programming concepts covered in the previous course • Review exercises and discussions to reinforce the foundational knowledge
Week 2-3	Week 2 - 3: Structure Compound Data Types • Introduction to structure data types in programming • Understanding how to define and use structures in C++ • Hands-on exercises to practice working with structures
Week 4-5	Week 4-5: String Manipulation • Exploring string data types and their manipulation in C++ • String functions and operations • Practical exercises and projects involving string manipulation
Week 6	Week 6: Dynamic Memory Allocation • Understanding dynamic memory allocation in C++ • Working with pointers and memory allocation functions (new, delete) • Practical examples and exercises to reinforce the concept
Week 7-8	Week 7-8: Files • Introduction to file handling in C++ • Reading from and writing to files • Exercises and projects involving file input/output operations

Week 9	Week 9: Midterm Examination • Midterm examination covering topics from weeks 1-8 • Review of previous topics and discussion of any questions or concerns
Week 10	Week 10: Exception Handling • Introduction to exception handling in C++ • Handling runtime errors and exceptional situations • Practice exercises and examples to understand exception handling mechanisms
Week 11-14	Week 11-14: Prepare Mini Project • Working on a mini project that integrates concepts learned so far • Planning, designing, and implementing a small-scale application or program • Regular progress check-ins and guidance throughout the project development
Week 15	Week 15: Revision and Review • Recap of all topics covered throughout the course • Review exercises, discussions, and Q&A sessions to solidify understanding • Final exam preparation and guidance

Learning and Teaching Resources مصادر التعلم والتدريس		
	Text	**Available in the Library?**
Required Texts	Problem Solving with C++ by Walter Savitch (Author), Kenrick Mock (Author)	
Recommended Texts		
Websites		

Grading Scheme مخطط الدرجات				
Group	**Grade**	التقدير	**Marks**	**Definition**

			%	
Success Group (50 - 100)	**A** - Excellent	امتياز	90 - 100	Outstanding Performance
	B - Very Good	جيد جدا	80 - 89	Above average with some errors
	C - Good	جيد	70 - 79	Sound work with notable errors
	D - Satisfactory	متوسط	60 - 69	Fair but with major shortcomings
	E - Sufficient	مقبول	50 - 59	Work meets minimum criteria
Fail Group (0 – 49)	**FX** – Fail	راسب (قيد المعالجة)	(45-49)	More work required but credit awarded
	F – Fail	راسب	(0-44)	Considerable amount of work required

Note: Marks Decimal places above or below 0.5 will be rounded to the higher or lower full mark (for example a mark of 54.5 will be rounded to 55, whereas a mark of 54.4 will be rounded to 54. The University has a policy NOT to condone "near-pass fails" so the only adjustment to marks awarded by the original marker(s) will be the automatic rounding outlined above.

MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information معلومات المادة الدراسية					
Module Title	**Computer Organization**			**Module Delivery**	
Module Type	**Core**			☒ **Theory** ☒ **Lecture** ☒ **Lab** ☐ **Tutorial** ☐ **Practical** ☐ **Seminar**	
Module Code	NT108				
ECTS Credits	**5**				
SWL (hr/sem)	**125**				
Module Level	1		**Semester of Delivery**	2	
Administering Department	Type Dept. Code		**College**	Type College Code	
Module Leader			**e-mail**		
Module Leader's Acad. Title	Lecturer		**Module Leader's Qualification**	Ph.D.	
Module Tutor	Name (if available)		**e-mail**	E-mail	
Peer Reviewer Name	Name		**e-mail**	E-mail	

Scientific Committee Approval Date		**Version Number**	1.0

Relation with other Modules العلاقة مع المواد الدراسية الأخرى			
Prerequisite module	NT104	**Semester**	
Co-requisites module	None	**Semester**	

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Objectives أهداف المادة الدراسية	1. Understanding Computer Architecture: o Gain a comprehensive understanding of computer organization and architecture principles. o Understand the components and their interconnections in a computer system. 2. Data Representation and Arithmetic: o Learn various number systems and their conversions (binary, decimal, hexadecimal). o Understand how data is represented and manipulated in a computer system. o Perform arithmetic operations on binary numbers. 3. Memory Systems: o Understand the organization and hierarchy of computer memory systems. o Learn about caching techniques and their impact on performance. o Study memory management and addressing techniques. 4. Instruction Set Architecture (ISA): o Learn about different instruction set architectures and their characteristics. o Understand the components and execution of instructions. o Analyze the relationship between ISA and machine language. 5. Processor Organization: o Understand the structure and components of a processor. o Learn about instruction pipelining and its benefits and challenges. o Study the design and implementation of control units. 6. Input/Output (I/O) Systems: o Learn about different I/O devices and their interfaces.

	○ Understand the techniques used for I/O data transfer. ○ Study interrupt handling and DMA (Direct Memory Access). 7. Parallel Processing and Multiprocessor Systems: ○ Understand the concepts of parallel processing and its benefits. ○ Study different parallel processing architectures and their characteristics. ○ Learn about multiprocessor systems and their organization. 8. Performance Evaluation and Optimization: ○ Learn performance metrics and evaluation techniques for computer systems. ○ Understand the factors affecting computer system performance. ○ Study optimization techniques to improve system performance. 9. Emerging Trends and Technologies: ○ Explore current and emerging trends in computer organization. ○ Study new technologies and their impact on computer systems. ○ Understand the challenges and opportunities in designing future computer architectures.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	1. Understand the fundamental principles of computer organization and architecture, including the components and their interactions within a computer system. 2. Demonstrate knowledge and proficiency in various number systems, data representation, and arithmetic operations in a computer system. 3. Analyze and evaluate different memory systems, including cache memory and main memory, and understand their impact on computer performance. 4. Describe and interpret different instruction set architectures (ISAs), including their components, instruction formats, and execution. 5. Analyze and evaluate the design and organization of processors, including pipelining techniques and control unit implementation. 6. Understand the principles and techniques of input/output (I/O) systems, including I/O devices, interfaces, interrupts, and direct memory access (DMA). 7. Discuss and evaluate parallel processing and multiprocessor systems, including concepts of parallelism, parallel architectures, and interconnectivity. 8. Apply performance evaluation techniques to measure and analyze the performance of computer systems, and propose optimization strategies for improving system performance. 9. Stay informed about current and emerging trends and technologies in computer organization, and assess their potential impact on future computer architectures. 10. Demonstrate effective problem-solving, critical thinking, and analytical skills in the context of computer organization and architecture. 11. Communicate effectively, both orally and in writing, about complex concepts and topics related to computer organization. **12.** Work collaboratively and contribute effectively as a team member in group projects and activities related to computer organization.
Indicative Contents	1. Introduction to Computer Organization:

المحتويات الإرشادية	○ Basic concepts and terminology in computer organization. ○ Historical development and evolution of computer architecture. 2. Digital Logic and Boolean Algebra: ○ Binary representation and arithmetic operations. ○ Logic gates, Boolean functions, and truth tables. ○ Combinational and sequential logic circuits. 3. Data Representation and Arithmetic: ○ Number systems: binary, decimal, hexadecimal. ○ Signed and unsigned integer representation. ○ Floating-point representation and arithmetic operations. 4. Central Processing Unit (CPU): ○ Instruction set architecture (ISA) and machine language. ○ CPU organization and components. ○ Control unit, instruction fetching, and execution. 5. Memory Systems: ○ Memory hierarchy and storage technologies. ○ Cache memory organization, principles, and mapping techniques. ○ Main memory organization and addressing modes. ○ Virtual memory concepts and techniques. 6. Input/Output (I/O) Systems: ○ I/O devices, interfaces, and data transfer methods. ○ Interrupt handling and interrupt-driven I/O. ○ Direct Memory Access (DMA) and its role in data transfer. 7. Pipeline Processing: ○ Instruction pipelining concepts and stages. ○ Hazards and techniques for hazard detection and resolution. ○ Performance metrics and improvements in pipeline processing. 8. Parallel Processing and Multiprocessor Systems: ○ Concepts of parallel processing and its benefits. ○ Types of parallel architectures: SIMD, MIMD, and multicore. ○ Interconnection networks and communication among processors. 9. Performance Evaluation and Optimization: ○ Performance metrics and measurement techniques. ○ Bottleneck identification and performance analysis. ○ Techniques for optimizing computer system performance. 10. Emerging Trends and Advanced Topics: ○ Advanced topics in computer organization, such as superscalar processors, out-of-order execution, and speculative execution. ○ Emerging technologies and their impact on computer organization, such as quantum computing and neuromorphic computing.

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	1. Understand the Fundamentals: ○ Start by grasping the foundational concepts and principles of

	computer organization, such as binary representation, digital logic, and Boolean algebra. o Build a strong understanding of number systems, data representation, and arithmetic operations used in computer systems. 2. Visualize and Diagram: o Use visual aids, diagrams, and flowcharts to represent and understand the structure and organization of computer components. o Draw diagrams to illustrate the flow of data and control signals within a computer system, such as the CPU, memory, and I/O devices. 3. Hands-on Experience: o Gain practical experience by working with computer hardware and software. This can involve assembling computers, configuring components, or writing low-level programs. o Experiment with simulators or emulators to observe how instructions are executed and how data flows through different computer components. 4. Relate to Real-World Examples: o Relate the concepts of computer organization to real-world examples and applications. Understand how the principles of computer organization are applied in everyday computing devices. 5. Analyze and Evaluate Case Studies: o Study and analyze case studies of actual computer architectures and designs. o Examine the trade-offs made in the design of different computer systems, considering factors such as performance, power consumption, and cost. 6. Solve Practice Problems: o Practice solving problems related to computer organization. This could involve analyzing and designing digital circuits, writing assembly language programs, or optimizing system performance. 7. Stay Updated with Current Research: o Keep up-to-date with the latest advancements and research in computer organization. o Read academic papers, attend conferences, and follow industry trends to understand emerging technologies and new approaches to computer organization. 8. Collaborate and Discuss: o Engage in discussions and collaborate with peers or study groups. Share knowledge, exchange ideas, and clarify concepts through group discussions or online forums. 9. Seek Guidance and Resources: o Consult textbooks, online resources, and academic materials that cover computer organization. o Seek guidance from instructors, tutors, or professionals with expertise in computer architecture and organization. 10. Practice Conceptual Mapping:

	○ Develop a conceptual map or framework to connect the different topics and components of computer organization. ○ Understand how the various concepts and components fit together to form a cohesive computer system.

Student Workload (SWL) الحمل الدراسي للطالب محسوب لـ ١٥ اسبوعا			
Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	63	**Structured SWL (h/w)** الحمل الدراسي المنتظم للطالب أسبوعيا	7
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	62	**Unstructured SWL (h/w)** الحمل الدراسي غير المنتظم للطالب أسبوعيا	6
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	125		

Module Evaluation
تقييم المادة الدراسية

		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	Quizzes	2	10% (10)	5 and 10	LO #1, #2 and #10, #11
	Assignments	2	10% (10)	2 and 12	LO #3, #4 and #6, #7
	Projects / Lab.	1	10% (10)	Continuous	All
	Report	1	10% (10)	13	LO #5, #8 and #10
Summative assessment	Midterm Exam	2hr	10% (10)	7	LO #1 - #7
	Final Exam	3hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus) المنهاج الاسبوعي النظري	
	Material Covered
Week 1	Introduction to Computer Organization, Overview of computer systems and their components (done)
Week 2	Digital Logic and Boolean Algebra,
Week 3	Data Representation and Arithmetic
Week 4	Central Processing Unit (CPU) (done)
Week 5	Instruction set architecture (ISA) and machine language
Week 6	CPU organization and components
Week 7	Control unit and instruction execution
Week 8	Memory Hierarchy (done)
Week 9	Memory organization and addressing (done)
Week 10	Cache memory: principles, levels, and mapping techniques (done)
Week 11	Input/Output Systems, Interrupts and DMA (Direct Memory Access), I/O performance and strategies
Week 12	Pipelining and Superscalar Techniques
Week 13	Multiprocessors and Parallel Computer Architecture
Week 14	Performance Evaluation and Benchmarking
Week 15	Review

Delivery Plan (Weekly Lab. Syllabus) المنهاج الاسبوعي للمختبر	
	Material Covered
Week 1	8086 system architecture
Week 2	8086 Instruction Set-1
Week 3	8086 Instruction Set-2
Week 4	8086 Instruction Set-3
Week 5	8086 Instruction Set-4
Week 6	8086 Instruction Set-5
Week 7	8086 Addressing Mode
Week 8	Memories (RAM, ROM) (done)
Week 9	Cache Memory ((done))
Week 10	8086 Programming Skills
Week 11	8086 Programming Skills
Week 12	8086 I/O unit
Week 13	Memory Mapped I/O, Isolated Input Output
Week 14	Memory/Input Output Interface
Week 15	Review

Learning and Teaching Resources

مصادر التعلم والتدريس		
	Text	**Available in the Library?**
Required Texts	Hwang K., 1993, "Advanced Computer Architecture: Parallelism ,Scalability and Programmability", *McGraw-Hill, Inc. ASIN: 7111067126.*	
Recommended Texts	Barry B. Brey, "The Intel Microprocessors: 8086/8088, 80186/80188, 80286, 80386, 80486, Pentium, and Pentium Pro Processor Architecture, Programming, and Interfacing", Pearson Education, 2010	
Websites	https://www.javatpoint.com/8086-microprocessor https://www.tutorialspoint.com/microprocessor/microprocessor_8086_functional_units.htm	

Grading Scheme مخطط الدرجات				
Group	**Grade**	التقدير	**Marks %**	**Definition**
Success Group (50 - 100)	**A** - Excellent	امتياز	90 - 100	Outstanding Performance
	B - Very Good	جيد جدا	80 - 89	Above average with some errors
	C - Good	جيد	70 - 79	Sound work with notable errors
	D - Satisfactory	متوسط	60 - 69	Fair but with major shortcomings
	E - Sufficient	مقبول	50 - 59	Work meets minimum criteria
Fail Group (0 – 49)	**FX** – Fail	راسب (قيد المعالجة)	(45-49)	More work required but credit awarded
	F – Fail	راسب	(0-44)	Considerable amount of work required

Note: Marks Decimal places above or below 0.5 will be rounded to the higher or lower full mark (for example a mark of 54.5 will be rounded to 55, whereas a mark of 54.4 will be rounded to 54. The University has a policy NOT to condone "near-pass fails" so the only adjustment to marks awarded by the original marker(s) will be the automatic rounding outlined above.

MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information

معلومات المادة الدراسية

Module Title	Data Communication and Networking	Module Delivery
Module Type	Core	☒ Theory ☒ Lecture ☐ Lab ☒ Tutorial ☐ Practical ☐ Seminar
Module Code	NT109	
ECTS Credits	5	
SWL (hr/sem)	125	

Module Level	1	**Semester of Delivery**	2
Administering Department	NT	**College**	CSM
Module Leader	Name	**e-mail**	E-mail
Module Leader's Acad. Title		**Module Leader's Qualification**	
Module Tutor	Name (if available)	**e-mail**	E-mail
Peer Reviewer Name	Name	**e-mail**	E-mail
Scientific Committee Approval Date		**Version Number**	1.0

Relation with other Modules

العلاقة مع المواد الدراسية الأخرى

Prerequisite module	NT101	**Semester**	
Co-requisites module	None	**Semester**	

Module Aims, Learning Outcomes and Indicative Contents

أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية

Module Objectives أهداف المادة الدراسية	In this course, Networks department aims to achieve the following objectives: 1. Understand the fundamental concepts and principles of data communication and networking. 2. Gain knowledge of communication systems and their components. 3. Familiarize with different communication network types and their characteristics. 4. Comprehend the protocol architecture and the OSI model. 5. Understand the functions and operations of each layer in the OSI model. 6. Gain knowledge of the physical layer, including digital and analog transmission and various transmission media.

	7. Understand data and signal concepts, analog, and digital signals, and transmission impairments. 8. Learn about digital transmission techniques, including digital-to-digital conversion and transmission modes. 9. Familiarize with analog transmission techniques, including analog-to-analog conversion and modulation. 10. Gain knowledge of multiplexing techniques such as FDM, TDM, and WDM, and understand synchronization methods. 11. Learn about guided transmission media, including twisted pair, coaxial cable, and fiber-optic. 12. Gain knowledge of unguided transmission media, including wireless, satellite, and microwave. 13. Understand error detection and correction techniques such as parity checking, checksum, and CRC. 14. Learn about multiplexing and multiple access techniques, including FDMA, TDMA, and CDMA. 15. Gain knowledge of wired LANs, with a focus on Ethernet standards, evolution, frame structure, and operation.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	Upon successful completion of the this course for the Networks department, students should be able to demonstrate the following learning outcomes: 1. Understand the fundamental concepts and principles of data communication and networking, including communication models, network types, and protocol architecture. 2. Demonstrate knowledge of the OSI model and its layers, and explain the functions and operations of each layer. 3. Describe the characteristics, advantages, and limitations of different transmission media, both guided and unguided. 4. Analyze and interpret data and signals, including analog and digital signals, and understand the factors affecting transmission quality. 5. Apply digital transmission techniques, including digital-to-digital conversion and various transmission modes. 6. Apply analog transmission techniques, including analog-to-analog conversion and modulation methods. 7. Demonstrate an understanding of multiplexing techniques, such as FDM, TDM, and WDM, and explain their advantages and applications. 8. Compare and contrast different guided transmission media, such as twisted pair, coaxial cable, and fiber-optic, based on their characteristics and capabilities. 9. Evaluate the characteristics and advantages of unguided transmission media, including wireless, satellite, and microwave technologies. 10. Apply error detection and correction techniques, including parity

	checking, checksum, and CRC, to ensure data integrity. 11. Analyze multiplexing and multiple access techniques, such as FDMA, TDMA, and CDMA, and understand their applications in communication systems. 12. Explain the Ethernet standard, its evolution, and the frame structure of Ethernet LANs. 13. Evaluate the security considerations and challenges associated with data communication and networking. 14. Demonstrate effective communication and collaboration skills in a networking context. 15. Apply theoretical knowledge to analyze and solve practical problems related to data communication and networking.
Indicative Contents المحتويات الإرشادية	The indicative contents of this course for the computer department may include the following topics: 1. Communication Systems • Introduction to communication systems • Communication models and components 2. Network Criteria and Communication Network Types • Network criteria (performance, reliability, security, etc.) • Communication network types 3. Protocol Architecture and OSI Model • Protocol architecture and layered approach • OSI model and its layers • Functions of each OSI layer 4. Physical Layer • Introduction to the physical layer • Digital and analog transmission • Transmission media: Guided and unguided 5. Data and Signals • Data and signal concepts • Analog and digital signals • Transmission impairments and noise 6. Digital-to-Digital Conversion • Digital-to-digital conversion techniques 7. Analog Transmission • Analog-to-analog conversion • Analog-to-digital conversion • Modulation techniques (AM, FM, PM) 8. Multiplexing • Multiplexing techniques (FDM, TDM, WDM) • Statistical multiplexing and its advantages

	• Synchronization and its types 9. Guided Transmission Media • Twisted pair, coaxial cable, and fiber-optic • Characteristics, advantages, and limitations 10. Unguided Transmission Media • Wireless, satellite, and microwave • Characteristics, advantages, and limitations 11. Error Detection and Correction • Introduction to error detection and correction • Parity checking, checksum, and CRC • Forward error correction techniques 12. Multiplexing and Multiple Access • Frequency division multiplexing (FDM) • Time division multiplexing (TDM) • Multiple access techniques (FDMA, TDMA, CDMA) 13. Wired LANs: Ethernet • Introduction to local area networks (LANs) • Ethernet standard and its evolution. • Ethernet frame structure and operation

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	Learning and teaching strategies for this course for the Network department can include a combination of the following: 39. Lectures: Engage students through informative lectures that cover theoretical concepts and provide an overview of key topics. Use multimedia resources, visuals, and real-world examples to enhance understanding. 40. Group Discussions and Collaborative Learning: Encourage group discussions and collaborative activities to foster interaction and knowledge sharing among students. Assign group projects or case studies that require teamwork and problem-solving. 41. Online Resources and Multimedia: Utilize online resources, interactive tutorials, and multimedia materials to supplement learning. This can include video lectures, online quizzes, virtual labs, and interactive modules. 42. Assignments and Projects: Assign individual and group projects that require students to apply their knowledge and skills to solve real-world problems or complete practical tasks. This promotes critical thinking, problem-solving, and practical application of concepts. 43. Assessments and Feedback: Conduct regular assessments, quizzes, and examinations to evaluate students' understanding of the course material.

	Provide timely and constructive feedback to help students identify areas of improvement. 44. Industry Visits and Field Trips: Organize visits to IT companies, data centers, or relevant organizations to expose students to real-world IT environments. This provides valuable industry insights and networking opportunities. 45. Online Discussion Forums and Communication Platforms: Establish online discussion forums or communication platforms where students can ask questions, share resources, and engage in discussions outside of the classroom. These strategies promote active learning, practical application of knowledge, and engagement with the subject matter. They cater to different learning styles and encourage students to develop critical thinking, problem-solving, and communication skills necessary for success in this field.

Student Workload (SWL)

الحمل الدراسي للطالب محسوب لـ ١٥ اسبوعا

Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	63	**Structured SWL (h/w)** الحمل الدراسي المنتظم للطالب أسبوعيا	6
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	62	**Unstructured SWL (h/w)** الحمل الدراسي غير المنتظم للطالب أسبوعيا	6
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	125		

Module Evaluation

تقييم المادة الدراسية

		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	**Quizzes**	2	10% (10)	5 and 10	LO #1, #2 and #10, #11
	Assignments	2	10% (10)	2 and 12	LO #3, #4 and #6, #7
	Projects / Lab.	1	10% (10)	Continuous	All
	Report	1	10% (10)	13	LO #5, #8 and #10
Summative assessment	**Midterm Exam**	2hr	10% (10)	7	LO #1 - #7
	Final Exam	3hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus) المنهاج الاسبوعي النظري	
	Material Covered
Week 1	• Communication Systems
Week 2	• Network Criteria and Communication Network Types
Week 3	• Protocol Architecture and OSI Model
Week 4	• Physical Layer
Week 5	• Data and Signals
Week 6	• Digital-to-Digital Conversion (Part 1)
Week 7	• Digital-to-Digital Conversion (Part 2)
Week 8	• Analog Transmission
Week 9	• Mid term
Week 10	• Multiplexing
Week 11	• Guided Transmission Media
Week 12	• Unguided Transmission Media
Week 13	• Error Detection and Correction
Week 14	• Wired LANs: Ethernet
Week 15	Week 15: Review
Week 16	**Preparatory week before the final Exam**

Learning and Teaching Resources مصادر التعلم والتدريس		
	Text	**Available in the Library?**
Required Texts	Data Communication and Networking	
Recommended Texts		

Websites	

Grading Scheme مخطط الدرجات				
Group	**Grade**	التقدير	**Marks %**	**Definition**
Success Group (50 - 100)	**A** - Excellent	امتياز	90 - 100	Outstanding Performance
	B - Very Good	جيد جدا	80 - 89	Above average with some errors
	C - Good	جيد	70 - 79	Sound work with notable errors
	D - Satisfactory	متوسط	60 - 69	Fair but with major shortcomings
	E - Sufficient	مقبول	50 - 59	Work meets minimum criteria
Fail Group (0 – 49)	**FX** – Fail	راسب (قيد المعالجة)	(45-49)	More work required but credit awarded
	F – Fail	راسب	(0-44)	Considerable amount of work required

Note: Marks Decimal places above or below 0.5 will be rounded to the higher or lower full mark (for example a mark of 54.5 will be rounded to 55, whereas a mark of 54.4 will be rounded to 54. The University has a policy NOT to condone "near-pass fails" so the only adjustment to marks awarded by the original marker(s) will be the automatic rounding outlined above.

MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information

معلومات المادة الدراسية

Module Title	**Probabilities and Statistics**	**Module Delivery**	
Module Type	**SUPPORT**	☒ **Theory** ☒ **Lecture** ☒ **Lab** ☐ **Tutorial** ☐ **Practical** ☐ **Seminar**	
Module Code	**NT110**		
ECTS Credits	**5**		
SWL (hr/sem)	**125**		

Module Level	1	**Semester of Delivery**	2
Administering Department	NT	**College**	CSM
Module Leader	Name	**e-mail**	E-mail
Module Leader's Acad. Title		**Module Leader's Qualification**	
Module Tutor	Name (if available)	**e-mail**	E-mail
Peer Reviewer Name	Name	**e-mail**	E-mail

Scientific Committee Approval Date		Version Number	1.0

Relation with other Modules العلاقة مع المواد الدراسية الأخرى			
Prerequisite module	None	**Semester**	
Co-requisites module	None	**Semester**	

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Objectives أهداف المادة الدراسية	The Probabilities and statistics basics course for the Networks department aims to achieve the following objectives: 1. Understand the fundamental concepts of probability theory: Students should develop a solid foundation in probability theory, including concepts such as sample spaces, events, probability axioms, conditional probability, and independence. 2. Apply probability concepts to real-world scenarios: Students should be able to apply probability theory to solve problems and analyze real-world situations that involve uncertainty and randomness. This includes calculating probabilities, understanding the concept of expected value, and analyzing random variables. 3. Comprehend statistical concepts and methods: Students should acquire a good understanding of statistical concepts, such as random variables, probability distributions, central tendency, variability, hypothesis testing, confidence intervals, and regression analysis. 4. Perform statistical data analysis: Students should learn how to collect, organize, and analyze data using appropriate statistical techniques. This includes techniques for data exploration, summarization, and inference. 5. Interpret and communicate statistical results: Students should be able to interpret the results of statistical analyses and effectively communicate their findings to others. This involves understanding the limitations of statistical methods and drawing appropriate conclusions from the data. 6. Apply statistical software: Students should gain hands-on experience

	with statistical software packages commonly used for data analysis, such as R, Python, or SPSS. They should be able to use these tools to perform statistical computations and generate graphical representations of data. 7. Develop critical thinking and problem-solving skills: The course aims to enhance students' critical thinking abilities by challenging them to analyze problems, evaluate evidence, and make informed decisions based on statistical reasoning. 8. Foster a strong mathematical foundation: Probability and Statistics often require a solid understanding of mathematical concepts, so the course aims to strengthen students' mathematical skills, including algebra, calculus, and basic mathematical notation. 9. Prepare for further study in related fields: The course may serve as a prerequisite or provide a foundation for more advanced courses in areas such as machine learning, data science, economics, psychology, or engineering, where probabilistic and statistical methods are commonly used.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	Upon successful completion of the Probabilities and statistics basics course for the Networks department, students should be able to demonstrate the following learning outcomes: 1. Understand fundamental probability concepts: Students should be able to demonstrate a strong understanding of basic probability concepts, including sample spaces, events, probability axioms, conditional probability, and independence. 2. Apply probability techniques: Students should be able to apply probability techniques to solve problems in various contexts, such as calculating probabilities of events, determining expected values, and understanding concepts like random variables and probability distributions. 3. Analyze statistical data: Students should be able to collect, organize, and analyze data using appropriate statistical methods. This includes understanding descriptive statistics, graphical representations of data, and basic inferential statistics. 4. Interpret statistical results: Students should be able to interpret the results of statistical analyses and draw meaningful conclusions. This involves understanding concepts such as confidence intervals, hypothesis testing, p-values, and statistical significance. 5. Apply statistical software: Students should be proficient in using statistical software packages (e.g., R, Python, SPSS) to perform data

	analysis and generate graphical representations of data. 6. Critically evaluate statistical claims: Students should be able to critically evaluate statistical claims and arguments presented in various contexts, such as scientific research, news articles, and advertisements. They should be able to identify common fallacies and recognize the importance of sound statistical reasoning. 7. Communicate statistical information: Students should be able to effectively communicate statistical information to both technical and non-technical audiences. This includes presenting findings, using appropriate visualizations, and conveying the limitations and implications of statistical analyses. 8. Apply statistical methods to real-world problems: Students should be able to apply their knowledge of probability and statistics to real-world problems in various fields, such as business, social sciences, engineering, or healthcare. They should be able to identify appropriate statistical methods and apply them to analyze and solve problems. 9. Develop critical thinking and problem-solving skills: The course should foster the development of critical thinking skills by engaging students in problem-solving activities that require them to think analytically, reason statistically, and make informed decisions based on data. 10. Prepare for further study or careers: The course should provide a solid foundation for students who wish to pursue further study or careers in fields that require a strong understanding of probability and statistics, such as data science, machine learning, economics, psychology, or research. These learning outcomes reflect the overarching goals of a Probabilities and Statistics course, which aim to equip students with the knowledge, skills, and tools necessary to understand and analyze data, make informed decisions, and apply statistical methods in various contexts.
Indicative Contents المحتويات الإرشادية	The indicative contents of the Probabilities and Statistics basics course for the computer department may include the following topics: 31. Introduction to Probability: o Basic concepts of probability: sample spaces, events, and outcomes. o Probability axioms and properties. o Combinatorics: permutations and combinations. o Conditional probability and independence. 32. Discrete Probability Distributions:

- Random variables and probability mass functions.
- Common discrete probability distributions: binomial, Poisson, and geometric distributions.
- Expected value and variance of discrete random variables.
- Joint probability distributions and conditional distributions.

33. Continuous Probability Distributions:

- Continuous random variables and probability density functions.
- Common continuous probability distributions: uniform, exponential, normal (Gaussian), and gamma distributions.
- Expected value and variance of continuous random variables.
- Joint probability distributions and conditional distributions.

34. Sampling and Data Description:

- Sampling techniques and sampling distributions.
- Descriptive statistics: measures of central tendency, measures of dispersion, and graphical representations of data.
- Data exploration and visualization.

35. Estimation and Confidence Intervals:

- Point estimation: methods for estimating population parameters.
- Interval estimation: construction and interpretation of confidence intervals.
- Sample size determination for estimation.

36. Hypothesis Testing:

- Null and alternative hypotheses.
- Test statistics and p-values.
- Types of errors and power of tests.
- Common hypothesis tests: z-tests, t-tests, chi-square tests.

37. Inference for Means and Proportions:

- Inference for population means: one-sample, independent samples, and paired samples.
- Inference for population proportions: one-sample and two-sample proportions.

38. Analysis of Variance (ANOVA):

- One-way ANOVA: comparing means of multiple groups.
- Post hoc tests and multiple comparisons.
- Two-way ANOVA: analyzing the effects of two factors.

39. Simple Linear Regression:

- The simple linear regression model.
- Least squares estimation and interpretation of coefficients.
- Assessing model fit and making predictions.

	40. Probability and Statistics in Decision Making: o Decision theory and utility. o Expected value and decision-making under uncertainty. o Risk assessment and risk management. 41. Introduction to Bayesian Statistics (optional): o Bayesian probability and Bayes' theorem. o Prior and posterior distributions. o Bayesian inference and decision-making. 42. Introduction to Statistical Software: o Hands-on experience with statistical software packages like R, Python, or SPSS. o Data manipulation, analysis, and visualization using software tools.

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	Learning and teaching strategies for the Probabilities and statistics basics course for the Network department can include a combination of the following: 1. Active Learning: Encourage active learning by incorporating activities that involve student participation, such as group discussions, problem-solving exercises, case studies, and hands-on data analysis projects. This approach helps students actively engage with the material, apply concepts, and develop a deeper understanding. 2. Real-World Examples: Use real-world examples and applications to demonstrate the relevance and practicality of probability and statistics. Relating the course content to everyday scenarios, industries, and research fields can enhance students' understanding and motivation. 3. Visual Representations: Utilize visual representations, such as charts, graphs, diagrams, and interactive simulations, to illustrate statistical concepts and relationships. Visual aids can help students visualize abstract concepts, interpret data, and identify patterns more effectively. 4. Technology Integration: Integrate statistical software tools, such as R, Python, or spreadsheet applications, into the course to facilitate data analysis and exploration. This hands-on experience with real-world data and statistical software enhances students' data manipulation and analysis skills.

5. Scaffolding: Break down complex topics into smaller, more manageable subtopics and provide scaffolding support to guide students through the learning process. Start with foundational concepts and gradually introduce more advanced topics, building upon prior knowledge.
6. Formative Assessment: Incorporate formative assessments, such as quizzes, in-class exercises, and homework assignments, to gauge students' understanding and provide feedback. This allows students to identify areas of weakness and reinforces learning throughout the course.
7. Problem-Based Learning: Present students with real-world problems or case studies that require the application of probability and statistical methods. This approach encourages critical thinking, problem-solving skills, and the integration of theoretical knowledge into practical scenarios.
8. Collaborative Learning: Promote collaboration and peer interaction through group activities, discussions, and projects. Working in teams allows students to learn from each other, share perspectives, and develop teamwork and communication skills.
9. Practical Exercises and Experiments: Incorporate practical exercises and experiments that involve collecting and analyzing data. This hands-on approach provides students with firsthand experience in data collection, manipulation, and statistical analysis, reinforcing theoretical concepts.
10. Reflection and Metacognition: Encourage students to reflect on their learning process and develop metacognitive skills. Regularly prompt students to evaluate their understanding, identify areas of improvement, and reflect on their learning strategies.
11. Office Hours and Support: Provide opportunities for individualized support, such as office hours or online discussion forums, where students can seek clarification, ask questions, and receive personalized guidance.
12. Engage with Resources: Encourage students to explore additional resources, such as textbooks, online tutorials, academic journals, or educational videos, to deepen their understanding and explore specific topics of interest.

By implementing these strategies, instructors can create an engaging and effective learning environment that fosters students' understanding, critical thinking skills, and practical application of probability and statistical concepts.

Student Workload (SWL)

الحمل الدراسي للطالب محسوب لـ ١٥ اسبوعا			
Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	63	**Structured SWL (h/w)** الحمل الدراسي المنتظم للطالب أسبوعيا	6
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	62	**Unstructured SWL (h/w)** الحمل الدراسي غير المنتظم للطالب أسبوعيا	6
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	125		

Module Evaluation
تقييم المادة الدراسية

		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	Quizzes	2	10% (10)	5 and 10	LO #1, #2 and #10, #11
	Assignments	2	10% (10)	2 and 12	LO #3, #4 and #6, #7
	Projects / Lab.	1	10% (10)	Continuous	All
	Report	1	10% (10)	13	LO #5, #8 and #10
Summative assessment	Midterm Exam	2hr	10% (10)	7	LO #1 - #7
	Final Exam	3hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)
المنهاج الاسبوعي النظري

	Material Covered
Week 1	Week 1: Introduction to Probabilities and statistics • Basic concepts of probability: sample spaces, events, and outcomes. • Probability axioms and properties. • Combinatorics: permutations and combinations. • Conditional probability and independence.
Week 2	Week 2: Introduction to Probabilities and statistics • Combinatorics: permutations and combinations. • Conditional probability and independence.

Week 3	Week 3: Discrete Probability Distributions • Random variables and probability mass functions. • Common discrete probability distributions: binomial, Poisson, and geometric distributions.
Week 4	Week 4: Discrete Probability Distributions • Expected value and variance of discrete random variables. • Joint probability distributions and conditional distributions.
Week 5	Week 5: Continuous Probability Distributions • Continuous random variables and probability density functions. • Common continuous probability distributions: uniform, exponential, normal (Gaussian), and gamma distributions.
Week 6	Week 6: Continuous Probability Distributions • Expected value and variance of continuous random variables. • Joint probability distributions and conditional distributions.
Week 7	Week 7: Sampling and Data Description • Sampling techniques and sampling distributions. • Descriptive statistics: measures of central tendency, measures of dispersion, and graphical representations of data.
Week 8	Week 8: Sampling and Data Description • Data exploration and visualization.
Week 9	Week 9: Estimation and Confidence Intervals • Point estimation: methods for estimating population parameters. • Interval estimation: construction and interpretation of confidence intervals.
Week 10	Week 10: Estimation and Confidence Intervals • Sample size determination for estimation
Week 11	Week 11: Hypothesis Testing • Null and alternative hypotheses. • Test statistics and p-values.

Week 12	Week 12: Hypothesis Testing • Types of errors and power of tests. • Common hypothesis tests: z-tests, t-tests, chi-square tests.
Week 13	Week 13: Inference for Means and Proportions • Inference for population means: one-sample, independent samples, and paired samples. • Inference for population proportions: one-sample and two-sample proportions.
Week 14	Week 14: Analysis of Variance (ANOVA) • One-way ANOVA: comparing means of multiple groups. • Post hoc tests and multiple comparisons. • Two-way ANOVA: analyzing the effects of two factors.
Week 15	Week 15: Review and Final Projects • Review of key concepts covered throughout the course • Completion of final projects or assignments demonstrating understanding of IT basics
Week 16	**Preparatory week before the final Exam**

Delivery Plan (Weekly Lab. Syllabus) المنهاج الاسبوعي للمختبر	
	Material Covered
Week 1	Introduction to R • Introduction to R environment and RStudio. • Basic R syntax, data types, and objects. • Reading data into R and basic data manipulation.
Week 2	Descriptive Statistics in R 2. Calculating measures of central tendency and dispersion. 3. Creating frequency tables and histograms. 4. Exploratory data analysis with R graphics.
Week 3	Probability Distributions in R 1. Generating random numbers from common probability distributions. 2. Calculating probabilities and percentiles.

	3. Plotting probability density functions and cumulative distribution functions.
Week 4	Sampling and Confidence Intervals in R 2. Simple random sampling in R. 3. Estimating population parameters and constructing confidence intervals. 4. Visualizing sampling distributions.
Week 5	Hypothesis Testing in R 2. Performing hypothesis tests for means and proportions. 3. Interpreting p-values and making decisions. 4. Conducting t-tests and chi-square tests in R.
Week 6	Analysis of Variance (ANOVA) in R 2. One-way ANOVA and post hoc tests. 3. Analyzing and interpreting ANOVA results. 4. Visualizing ANOVA data with boxplots and interaction plots.
Week 7	Simple Linear Regression in R • Fitting a simple linear regression model. • Assessing model fit and interpreting coefficients. • Predicting outcomes and evaluating the model.
Week 8	Multiple Linear Regression in R • Extending the simple linear regression to multiple predictors. • Model diagnostics and interpretation of results. Handling categorical predictors and interactions
Week 9	Logistic Regression in R • Introduction to logistic regression. • Fitting logistic regression models and interpreting coefficients. • Model assessment and prediction.
Week 10	Time Series Analysis in R • Introduction to time series data. • Time series decomposition and forecasting. • Analyzing and visualizing time series data.

Week 11	: Nonparametric Methods in R • Wilcoxon rank-sum test and Wilcoxon signed-rank test. • Kruskal-Wallis test and Friedman test. • Conducting nonparametric tests in R.
Week 12	Bayesian Statistics in R (optional) • Introduction to Bayesian inference. • Fitting Bayesian models and sampling from posterior distributions. • Interpreting and comparing Bayesian results.
Week 13	: Data Analysis Projects • Students work on data analysis projects applying concepts and techniques learned throughout the course. • Guidance, support, and feedback provided by the instructor during lab sessions.
Week 14	: Data Analysis Projects • Students work on data analysis projects applying concepts and techniques learned throughout the course. • Guidance, support, and feedback provided by the instructor during lab sessions.
Week 15	: Review and Wrap-up • Recap of key concepts and techniques covered throughout the course. • Q&A sessions, review exercises, and additional practice.

Learning and Teaching Resources
مصادر التعلم والتدريس

	Text	Available in the Library?
Required Texts		
Recommended Texts		
Websites		

Grading Scheme
مخطط الدرجات

Group	Grade	التقدير	Marks %	Definition
Success Group (50 - 100)	**A** - Excellent	امتياز	90 - 100	Outstanding Performance
	B - Very Good	جيد جدا	80 - 89	Above average with some errors
	C - Good	جيد	70 - 79	Sound work with notable errors
	D - Satisfactory	متوسط	60 - 69	Fair but with major shortcomings
	E - Sufficient	مقبول	50 - 59	Work meets minimum criteria
Fail Group (0 – 49)	**FX** – Fail	راسب (قيد المعالجة)	(45-49)	More work required but credit awarded
	F – Fail	راسب	(0-44)	Considerable amount of work required

Note: Marks Decimal places above or below 0.5 will be rounded to the higher or lower full mark (for example a mark of 54.5 will be rounded to 55, whereas a mark of 54.4 will be rounded to 54. The University has a policy NOT to condone "near-pass fails" so the only adjustment to marks awarded by the original marker(s) will be the automatic rounding outlined above.

MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information

معلومات المادة الدراسية

Module Title	**Probabilities and Statistics**	**Module Delivery**	
Module Type	**SUPPORT**	☒ **Theory** ☒ **Lecture** ☒ **Lab** ☐ **Tutorial** ☐ **Practical** ☐ **Seminar**	
Module Code	**NT110**		
ECTS Credits	**5**		
SWL (hr/sem)	**125**		

Module Level	1	**Semester of Delivery**	2
Administering Department	NT	**College**	CSM
Module Leader	Name	**e-mail**	E-mail
Module Leader's Acad. Title		**Module Leader's Qualification**	
Module Tutor	Name (if available)	**e-mail**	E-mail
Peer Reviewer Name	Name	**e-mail**	E-mail
Scientific Committee Approval Date		**Version Number**	1.0

Relation with other Modules العلاقة مع المواد الدراسية الأخرى			
Prerequisite module	None	**Semester**	
Co-requisites module	None	**Semester**	

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Objectives أهداف المادة الدراسية	The Probabilities and statistics basics course for the Networks department aims to achieve the following objectives: 10. Understand the fundamental concepts of probability theory: Students should develop a solid foundation in probability theory, including concepts such as sample spaces, events, probability axioms, conditional probability, and independence. 11. Apply probability concepts to real-world scenarios: Students should be able to apply probability theory to solve problems and analyze real-world situations that involve uncertainty and randomness. This includes calculating probabilities, understanding the concept of expected value, and analyzing random variables. 12. Comprehend statistical concepts and methods: Students should acquire a good understanding of statistical concepts, such as random variables, probability distributions, central tendency, variability, hypothesis testing, confidence intervals, and regression analysis. 13. Perform statistical data analysis: Students should learn how to collect, organize, and analyze data using appropriate statistical techniques. This includes techniques for data exploration, summarization, and inference. 14. Interpret and communicate statistical results: Students should be able to interpret the results of statistical analyses and effectively communicate their findings to others. This involves understanding the limitations of statistical methods and drawing appropriate conclusions from the data. 15. Apply statistical software: Students should gain hands-on experience with statistical software packages commonly used for data analysis, such as R, Python, or SPSS. They should be able to use these

	tools to perform statistical computations and generate graphical representations of data. 16. Develop critical thinking and problem-solving skills: The course aims to enhance students' critical thinking abilities by challenging them to analyze problems, evaluate evidence, and make informed decisions based on statistical reasoning. 17. Foster a strong mathematical foundation: Probability and Statistics often require a solid understanding of mathematical concepts, so the course aims to strengthen students' mathematical skills, including algebra, calculus, and basic mathematical notation. 18. Prepare for further study in related fields: The course may serve as a prerequisite or provide a foundation for more advanced courses in areas such as machine learning, data science, economics, psychology, or engineering, where probabilistic and statistical methods are commonly used.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	Upon successful completion of the Probabilities and statistics basics course for the Networks department, students should be able to demonstrate the following learning outcomes: 11. Understand fundamental probability concepts: Students should be able to demonstrate a strong understanding of basic probability concepts, including sample spaces, events, probability axioms, conditional probability, and independence. 12. Apply probability techniques: Students should be able to apply probability techniques to solve problems in various contexts, such as calculating probabilities of events, determining expected values, and understanding concepts like random variables and probability distributions. 13. Analyze statistical data: Students should be able to collect, organize, and analyze data using appropriate statistical methods. This includes understanding descriptive statistics, graphical representations of data, and basic inferential statistics. 14. Interpret statistical results: Students should be able to interpret the results of statistical analyses and draw meaningful conclusions. This involves understanding concepts such as confidence intervals, hypothesis testing, p-values, and statistical significance. 15. Apply statistical software: Students should be proficient in using statistical software packages (e.g., R, Python, SPSS) to perform data analysis and generate graphical representations of data. 16. Critically evaluate statistical claims: Students should be able to

	critically evaluate statistical claims and arguments presented in various contexts, such as scientific research, news articles, and advertisements. They should be able to identify common fallacies and recognize the importance of sound statistical reasoning. 17. Communicate statistical information: Students should be able to effectively communicate statistical information to both technical and non-technical audiences. This includes presenting findings, using appropriate visualizations, and conveying the limitations and implications of statistical analyses. 18. Apply statistical methods to real-world problems: Students should be able to apply their knowledge of probability and statistics to real-world problems in various fields, such as business, social sciences, engineering, or healthcare. They should be able to identify appropriate statistical methods and apply them to analyze and solve problems. 19. Develop critical thinking and problem-solving skills: The course should foster the development of critical thinking skills by engaging students in problem-solving activities that require them to think analytically, reason statistically, and make informed decisions based on data. 20. Prepare for further study or careers: The course should provide a solid foundation for students who wish to pursue further study or careers in fields that require a strong understanding of probability and statistics, such as data science, machine learning, economics, psychology, or research. These learning outcomes reflect the overarching goals of a Probabilities and Statistics course, which aim to equip students with the knowledge, skills, and tools necessary to understand and analyze data, make informed decisions, and apply statistical methods in various contexts.
Indicative Contents المحتويات الإرشادية	The indicative contents of the Probabilities and Statistics basics course for the computer department may include the following topics: 43. Introduction to Probability: o Basic concepts of probability: sample spaces, events, and outcomes. o Probability axioms and properties. o Combinatorics: permutations and combinations. o Conditional probability and independence. 44. Discrete Probability Distributions: o Random variables and probability mass functions.

- Common discrete probability distributions: binomial, Poisson, and geometric distributions.
- Expected value and variance of discrete random variables.
- Joint probability distributions and conditional distributions.

45. Continuous Probability Distributions:

- Continuous random variables and probability density functions.
- Common continuous probability distributions: uniform, exponential, normal (Gaussian), and gamma distributions.
- Expected value and variance of continuous random variables.
- Joint probability distributions and conditional distributions.

46. Sampling and Data Description:

- Sampling techniques and sampling distributions.
- Descriptive statistics: measures of central tendency, measures of dispersion, and graphical representations of data.
- Data exploration and visualization.

47. Estimation and Confidence Intervals:

- Point estimation: methods for estimating population parameters.
- Interval estimation: construction and interpretation of confidence intervals.
- Sample size determination for estimation.

48. Hypothesis Testing:

- Null and alternative hypotheses.
- Test statistics and p-values.
- Types of errors and power of tests.
- Common hypothesis tests: z-tests, t-tests, chi-square tests.

49. Inference for Means and Proportions:

- Inference for population means: one-sample, independent samples, and paired samples.
- Inference for population proportions: one-sample and two-sample proportions.

50. Analysis of Variance (ANOVA):

- One-way ANOVA: comparing means of multiple groups.
- Post hoc tests and multiple comparisons.
- Two-way ANOVA: analyzing the effects of two factors.

51. Simple Linear Regression:

- The simple linear regression model.
- Least squares estimation and interpretation of coefficients.
- Assessing model fit and making predictions.

52. Probability and Statistics in Decision Making:

	○ Decision theory and utility. ○ Expected value and decision-making under uncertainty. ○ Risk assessment and risk management. 53. Introduction to Bayesian Statistics (optional): ○ Bayesian probability and Bayes' theorem. ○ Prior and posterior distributions. ○ Bayesian inference and decision-making. 54. Introduction to Statistical Software: ○ Hands-on experience with statistical software packages like R, Python, or SPSS. ○ Data manipulation, analysis, and visualization using software tools.

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	Learning and teaching strategies for the Probabilities and statistics basics course for the Network department can include a combination of the following: 13. Active Learning: Encourage active learning by incorporating activities that involve student participation, such as group discussions, problem-solving exercises, case studies, and hands-on data analysis projects. This approach helps students actively engage with the material, apply concepts, and develop a deeper understanding. 14. Real-World Examples: Use real-world examples and applications to demonstrate the relevance and practicality of probability and statistics. Relating the course content to everyday scenarios, industries, and research fields can enhance students' understanding and motivation. 15. Visual Representations: Utilize visual representations, such as charts, graphs, diagrams, and interactive simulations, to illustrate statistical concepts and relationships. Visual aids can help students visualize abstract concepts, interpret data, and identify patterns more effectively. 16. Technology Integration: Integrate statistical software tools, such as R, Python, or spreadsheet applications, into the course to facilitate data analysis and exploration. This hands-on experience with real-world data and statistical software enhances students' data manipulation and analysis skills.

17. Scaffolding: Break down complex topics into smaller, more manageable subtopics and provide scaffolding support to guide students through the learning process. Start with foundational concepts and gradually introduce more advanced topics, building upon prior knowledge.

18. Formative Assessment: Incorporate formative assessments, such as quizzes, in-class exercises, and homework assignments, to gauge students' understanding and provide feedback. This allows students to identify areas of weakness and reinforces learning throughout the course.

19. Problem-Based Learning: Present students with real-world problems or case studies that require the application of probability and statistical methods. This approach encourages critical thinking, problem-solving skills, and the integration of theoretical knowledge into practical scenarios.

20. Collaborative Learning: Promote collaboration and peer interaction through group activities, discussions, and projects. Working in teams allows students to learn from each other, share perspectives, and develop teamwork and communication skills.

21. Practical Exercises and Experiments: Incorporate practical exercises and experiments that involve collecting and analyzing data. This hands-on approach provides students with firsthand experience in data collection, manipulation, and statistical analysis, reinforcing theoretical concepts.

22. Reflection and Metacognition: Encourage students to reflect on their learning process and develop metacognitive skills. Regularly prompt students to evaluate their understanding, identify areas of improvement, and reflect on their learning strategies.

23. Office Hours and Support: Provide opportunities for individualized support, such as office hours or online discussion forums, where students can seek clarification, ask questions, and receive personalized guidance.

24. Engage with Resources: Encourage students to explore additional resources, such as textbooks, online tutorials, academic journals, or educational videos, to deepen their understanding and explore specific topics of interest.

By implementing these strategies, instructors can create an engaging and effective learning environment that fosters students' understanding, critical thinking skills, and practical application of probability and statistical concepts.

Student Workload (SWL) الحمل الدراسي للطالب محسوب لـ ١٥ اسبوعا			
Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	63	**Structured SWL (h/w)** الحمل الدراسي المنتظم للطالب أسبوعيا	6
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	62	**Unstructured SWL (h/w)** الحمل الدراسي غير المنتظم للطالب أسبوعيا	6
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	125		

Module Evaluation

تقييم المادة الدراسية

		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	**Quizzes**	2	10% (10)	5 and 10	LO #1, #2 and #10, #11
	Assignments	2	10% (10)	2 and 12	LO #3, #4 and #6, #7
	Projects / Lab.	1	10% (10)	Continuous	All
	Report	1	10% (10)	13	LO #5, #8 and #10
Summative assessment	**Midterm Exam**	2hr	10% (10)	7	LO #1 - #7
	Final Exam	3hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)

المنهاج الاسبوعي النظري

	Material Covered
Week 1	Week 1: Introduction to Probabilities and statistics • Basic concepts of probability: sample spaces, events, and outcomes. • Probability axioms and properties. • Combinatorics: permutations and combinations. • Conditional probability and independence.
Week 2	Week 2: Introduction to Probabilities and statistics • Combinatorics: permutations and combinations.

	• Conditional probability and independence.
Week 3	Week 3: Discrete Probability Distributions • Random variables and probability mass functions. • Common discrete probability distributions: binomial, Poisson, and geometric distributions.
Week 4	Week 4: Discrete Probability Distributions • Expected value and variance of discrete random variables. • Joint probability distributions and conditional distributions.
Week 5	Week 5: Continuous Probability Distributions • Continuous random variables and probability density functions. • Common continuous probability distributions: uniform, exponential, normal (Gaussian), and gamma distributions.
Week 6	Week 6: Continuous Probability Distributions • Expected value and variance of continuous random variables. • Joint probability distributions and conditional distributions.
Week 7	Week 7: Sampling and Data Description • Sampling techniques and sampling distributions. • Descriptive statistics: measures of central tendency, measures of dispersion, and graphical representations of data.
Week 8	Week 8: Sampling and Data Description • Data exploration and visualization.
Week 9	Week 9: Estimation and Confidence Intervals • Point estimation: methods for estimating population parameters. • Interval estimation: construction and interpretation of confidence intervals.
Week 10	Week 10: Estimation and Confidence Intervals • Sample size determination for estimation
Week 11	Week 11: Hypothesis Testing • Null and alternative hypotheses.

	• Test statistics and p-values.
Week 12	Week 12: Hypothesis Testing • Types of errors and power of tests. • Common hypothesis tests: z-tests, t-tests, chi-square tests.
Week 13	Week 13: Inference for Means and Proportions • Inference for population means: one-sample, independent samples, and paired samples. • Inference for population proportions: one-sample and two-sample proportions.
Week 14	Week 14: Analysis of Variance (ANOVA) • One-way ANOVA: comparing means of multiple groups. • Post hoc tests and multiple comparisons. • Two-way ANOVA: analyzing the effects of two factors.
Week 15	Week 15: Review and Final Projects • Review of key concepts covered throughout the course • Completion of final projects or assignments demonstrating understanding of IT basics
Week 16	**Preparatory week before the final Exam**

Delivery Plan (Weekly Lab. Syllabus) المنهاج الاسبوعي للمختبر	
	Material Covered
Week 1	Introduction to R • Introduction to R environment and RStudio. • Basic R syntax, data types, and objects. • Reading data into R and basic data manipulation.
Week 2	Descriptive Statistics in R 5. Calculating measures of central tendency and dispersion. 6. Creating frequency tables and histograms. 7. Exploratory data analysis with R graphics.
Week 3	Probability Distributions in R 4. Generating random numbers from common probability distributions.

	5. Calculating probabilities and percentiles. 6. Plotting probability density functions and cumulative distribution functions.
Week 4	Sampling and Confidence Intervals in R 5. Simple random sampling in R. 6. Estimating population parameters and constructing confidence intervals. 7. Visualizing sampling distributions.
Week 5	Hypothesis Testing in R 5. Performing hypothesis tests for means and proportions. 6. Interpreting p-values and making decisions. 7. Conducting t-tests and chi-square tests in R.
Week 6	Analysis of Variance (ANOVA) in R 5. One-way ANOVA and post hoc tests. 6. Analyzing and interpreting ANOVA results. 7. Visualizing ANOVA data with boxplots and interaction plots.
Week 7	Simple Linear Regression in R • Fitting a simple linear regression model. • Assessing model fit and interpreting coefficients. • Predicting outcomes and evaluating the model.
Week 8	Multiple Linear Regression in R • Extending the simple linear regression to multiple predictors. • Model diagnostics and interpretation of results. Handling categorical predictors and interactions
Week 9	Logistic Regression in R • Introduction to logistic regression. • Fitting logistic regression models and interpreting coefficients. • Model assessment and prediction.
Week 10	Time Series Analysis in R • Introduction to time series data. • Time series decomposition and forecasting. • Analyzing and visualizing time series data.

Week 11	: Nonparametric Methods in R • Wilcoxon rank-sum test and Wilcoxon signed-rank test. • Kruskal-Wallis test and Friedman test. • Conducting nonparametric tests in R.
Week 12	Bayesian Statistics in R (optional) • Introduction to Bayesian inference. • Fitting Bayesian models and sampling from posterior distributions. • Interpreting and comparing Bayesian results.
Week 13	: Data Analysis Projects • Students work on data analysis projects applying concepts and techniques learned throughout the course. • Guidance, support, and feedback provided by the instructor during lab sessions.
Week 14	: Data Analysis Projects • Students work on data analysis projects applying concepts and techniques learned throughout the course. • Guidance, support, and feedback provided by the instructor during lab sessions.
Week 15	: Review and Wrap-up • Recap of key concepts and techniques covered throughout the course. • Q&A sessions, review exercises, and additional practice.

Learning and Teaching Resources
مصادر التعلم والتدريس

	Text	Available in the Library?
Required Texts		
Recommended Texts		
Websites		

Grading Scheme

مخطط الدرجات				
Group	**Grade**	التقدير	**Marks %**	**Definition**
Success Group (50 - 100)	**A** - Excellent	امتياز	90 - 100	Outstanding Performance
	B - Very Good	جيد جدا	80 - 89	Above average with some errors
	C - Good	جيد	70 - 79	Sound work with notable errors
	D - Satisfactory	متوسط	60 - 69	Fair but with major shortcomings
	E - Sufficient	مقبول	50 - 59	Work meets minimum criteria
Fail Group (0 – 49)	**FX** – Fail	راسب (قيد المعالجة)	(45-49)	More work required but credit awarded
	F – Fail	راسب	(0-44)	Considerable amount of work required

Note: Marks Decimal places above or below 0.5 will be rounded to the higher or lower full mark (for example a mark of 54.5 will be rounded to 55, whereas a mark of 54.4 will be rounded to 54. The University has a policy NOT to condone "near-pass fails" so the only adjustment to marks awarded by the original marker(s) will be the automatic rounding outlined above.

MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information معلومات المادة الدراسية					
Module Title	**Arabic Language**			**Module Delivery**	
Module Type	**Support**			☒ **Theory** ☒ **Lecture** ☐ **Lab** ☒ **Tutorial** ☐ **Practical** ☐ **Seminar**	
Module Code	**UOM101**				
ECTS Credits	**2**				
SWL (hr/sem)	**50**				
Module Level	**1**		**Semester of Delivery**	2	
Administering Department	**NT**		**College**	CSM	
Module Leader			**e-mail**		
Module Leader's Acad. Title			**Module Leader's Qualification**	Ph.D.	
Module Tutor			**e-mail**		
Peer Reviewer Name			**e-mail**		
Scientific Committee Approval Date	18/06/2023		**Version Number**	1.0	

Relation with other Modules العلاقة مع المواد الدراسية الأخرى			
Prerequisite module	None	**Semester**	
Co-requisites module	None	**Semester**	

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Objectives أهداف المادة الدراسية	1- التعرف على الكلام العربي: من ناحية تعريفة، اقسامة، الى علامات كل قسم منه. 2- معرفة الجملة العربية واقسام الجملة العربية والجمل الاسمية والجمل الفعلية 3- التعرف على حركات الاعراب: سواء كانت اصلية او فرعية 4- معرفة الطالب بالعفل العربي: من حيث الصحة والاعلال 5- معرفة الطالب الفعل العربي من حيث اللزوم والتعدي 6- معرفة لطالب الفعل العربي من حيث الزمن 7- طرق كتابة العدد و تذكرة وتانيثه 8- معرفة علامات الترقيم في الكلام 9- تعلم قواعد رسم الهمزة 10- التعرف على طريقة كتابة التاء المربوطة، والمبسوطة 11- قل ولا تقل: الأخطاء الشائعة لدى المتكلمين والكتاب 12- معرفة ماهو الأسلوب الخبري، 13- معرفة ماهو الأسلوب الانشائي، 14- تعلم مهارات لغوية: تنمية الذوق اللغوي، وتحسين الأسلوب لدى المتعلمين
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	1- ان يعرف الطالب الكلام العربي: من ناحية تعريفة، اقسامة، الى علامات كل قسم منه. 2- ان يتعلم الطالب الجملة العربية واقسام الجملة العربية والجمل الاسمية والجمل الفعلية 3- التعرف على حركات الاعراب: سواء كانت اصلية او فرعية 4- ان يعرف الطالب العفل العربي: من حيث الصحة والاعلال 5- ان يتعلم الطالب الفعل العربي من حيث اللزوم والتعدي 6- معرفة الطالب الفعل العربي من حيث الزمن 7- معرف الطالب طرق كتابة العدد و تذكرة وتانيثه 8- معرفة الطالب لعلامات الترقيم في الكلام 9- ان يتعلم الطالب قواعد رسم الهمزة 10- معرف الطالب على طريقة كتابة التاء المربوطة، والمبسوطة 11- قل ولا تقل: الأخطاء الشائعة لدى المتكلمين والكتاب 12- التعرف على الأسلوب الخبري، 13- معرفة ماهو الأسلوب الانشائي، 14- التعلم على مهارات لغوية: تنمية الذوق اللغوي، وتحسين الأسلوب لدى المتعلمين
Indicative Contents	1- التعرف على الكلام العربي: من ناحية تعريفة، اقسامة، الى علامات كل قسم منه[ساعه 2]

المحتويات الإرشادية	-2 معرفة الجملة العربية واقسام الجملة العربية والجمل الاسمية والجمل الفعلية، ساعه 2 -3 التعرف على حركات الاعراب: سواء كانت اصلية او فرعية، ساعه 2 -4 معرفة الطالب بالعفل العربي: من حيث الصحة والاعلال، ساعه 2 -5 معرفة الطالب الفعل العربي من حيث اللزوم والتعدي، ساعه 2 -6 معرفة لطالب الفعل العربي من حيث الزمن، ساعه 2 -7 طرق كتابة العدد و تذكرة وتانيثه، ساعه 2 -8 معرفة علامات الترقيم في الكلام، ساعه 2 -9 تعلم قواعد رسم الهمزة، ساعه 2 -10 التعرف على طريقة كتابة التاء المربوطة، والمبسوطة، ساعه 2 -11 قل ولا تقل: الأخطاء الشائعة لدى المتكلمين والكتاب ، ساعه 2 -12 معرفة ماهو الأسلوب الخبري، ساعه 2 -13 معرفة ماهو الأسلوب الانشائي، ساعه 2 -14 التعلم مهارات لغوية: تنمية الذوق اللغوي، وتحسين الأسلوب لدى المتعلمين، ساعه 2

Learning and Teaching Strategies
استراتيجيات التعلم والتعليم

Strategies	الإستراتيجية الرئيسية التي سيتم تبنيها في تقديم هذه الوحدة هي تشجيع الطلاب على المشاركة على المشاركه في الكلام الفربي وكتابته بالصورة الصحيحه ، مع تحسين مهارات التفكير النقدي وتوسيعها في نفس الوقت. سيتم تحقيق ذلك من خلال الفصول والبرامج التعليمية التفاعلية ومن خلال النظر في أنواع التجارب البسيطة التي تتضمن بعض أنشطة أخذ العينات التي تهم الطلاب.

Student Workload (SWL)
الحمل الدراسي للطالب محسوب لـ ١٥ اسبوعا

Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	32	**Structured SWL (h/w)** الحمل الدراسي المنتظم للطالب أسبوعيا	1
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	18	**Unstructured SWL (h/w)** الحمل الدراسي غير المنتظم للطالب أسبوعيا	1
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	50		

Module Evaluation
تقييم المادة الدراسية

		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative	**Quizzes**	3	15% (15)	5 and 10	LO #1, #2 and #10, #11

assessment	**Assignments**	3	15% (15)	2 and 12	LO #3, #4 and #6, #7
	Projects / Lab.				
	Report	1	10% (10)	13	LO #5, #8 and #10
Summative assessment	**Midterm Exam**	2hr	10% (10)	7	LO #1 - #7
	Final Exam	3hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus) المنهاج الاسبوعي النظري	
	Material Covered
Week 1	الكلام العربي: تعريفة، اقسامة، وعلامات كل قسم.
Week 2	الجملة العربية: تعريفها ، اقسامها : الاسمية والفعلية
Week 3	حركات الاعراب: اصلية، فرعية
Week 4	العفل العربي: من حيث الصحة والاعلال
Week 5	الفعل العربي من حيث اللزوم والتعدي
Week 6	الفعل العربي من حيث الزم
Week 7	امتحان
Week 8	العدد: تذكرة، وتانيثه
Week 9	علامات الترقيم في الكلام
Week 10	قواعد رسم الهمزة
Week 11	التاء المربوطة، والمبسوطة
Week 12	قل ولا تقل: الأخطاء الشائعة لدى المتكلمين والكتاب
Week 13	الأسلوب الخبري،
Week 14	والأسلوب الإنشائي
Week 15	مهارات لغوية: تنمية الذوق اللغوي، وتحسين الأسلوب لدى المتعلمين
Week 16	امتحان نهاية الفصل

Delivery Plan (Weekly Lab. Syllabus) المنهاج الاسبوعي للمختبر	
	Material Covered

Week 1	Lab 1:
Week 2	Lab 2
Week 3	Lab 3
Week 4	Lab 4:
Week 5	Lab 5:
Week 6	Lab 6:
Week 7	Lab 7:

Learning and Teaching Resources مصادر التعلم والتدريس		
	Text	**Available in the Library?**
Required Texts	جامع الدروس العربية: الشيخ مصطفى الغلاييني	No
Recommended Texts	الجملة العربية: تأليفها وأقسامها د. فاضل السامرائي	No
Websites	https://www.almrsal.com/post/923401	

Grading Scheme مخطط الدرجات				
Group	**Grade**	التقدير	**Marks %**	**Definition**
Success Group (50 - 100)	**A -** Excellent	امتياز	90 - 100	Outstanding Performance
	B - Very Good	جيد جدا	80 - 89	Above average with some errors
	C - Good	جيد	70 - 79	Sound work with notable errors
	D - Satisfactory	متوسط	60 - 69	Fair but with major shortcomings
	E - Sufficient	مقبول	50 - 59	Work meets minimum criteria
Fail Group (0 – 49)	**FX –** Fail	راسب (قيد المعالجة)	(45-49)	More work required but credit awarded
	F – Fail	راسب	(0-44)	Considerable amount of work required

Note: Marks Decimal places above or below 0.5 will be rounded to the higher or lower full mark (for example a mark of 54.5 will be rounded to 55, whereas a mark of 54.4 will be rounded to 54. The University has a policy NOT to condone "near-pass fails" so the only adjustment to marks awarded by the original marker(s) will be the automatic rounding outlined above.

MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information

معلومات المادة الدراسية

Module Title	**Data Structures**	**Module Delivery**
Module Type	**Core**	☒ **Theory** ☒ **Lecture** ☒ **Lab** ☐ **Tutorial** ☐ **Practical** ☐ **Seminar**
Module Code		

Module Aims, Learning Outcomes and Indicative Contents

أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية

Module Objectives

أهداف المادة الدراسية

The objectives of the "Data Structures" course are:

1. Understanding Fundamental Data Structures: To introduce students to the fundamental data structures such as arrays, linked lists, stacks, queues, trees, and graphs, and their characteristics, operations, and applications.
2. Implementing Data Structures: To provide students with hands-on experience in implementing data structures using programming languages, allowing them to understand the internal workings and mechanisms of these structures.
3. Algorithmic Problem-Solving: To develop students' problem-solving skills by designing and implementing algorithms using appropriate data structures, and to apply these algorithms to solve real-world problems efficiently.
4. Performance Optimization: To teach students techniques for optimizing the performance of data structures and algorithms, such as choosing the most suitable data structure for a given problem and employing efficient algorithms for common operations.
5. Teamwork and Collaboration: To encourage teamwork and collaboration through group projects and assignments, enabling students to work effectively in teams and learn from each other's perspectives and approaches.
6. Critical Thinking and Analysis: To foster critical thinking and analytical skills by challenging students to evaluate, modify, and improve existing data structures and algorithms, and to adapt them to new problem scenarios.
7. Practical Application: To demonstrate the practical application of data structures in various domains such as software development, database management, networking, and artificial intelligence, emphasizing their relevance in real-world scenarios.

By achieving these objectives, students will develop a strong foundation in data structures and acquire the skills necessary to design, implement, and analyze efficient algorithms and data structures for solving complex problems

Module Learning Outcomes مخرجات التعلم للمادة الدراسية	Upon completion of the "Data Structures" course, students will be able to: 1. Understand and Identify Data Structures: Identify and differentiate various data structures such as arrays, linked lists, stacks, queues, trees, and graphs, and understand their characteristics, advantages, and limitations. 2. Implement Data Structures: Implement data structures using programming languages, demonstrating proficiency in coding and understanding the internal workings and mechanisms of data structures. 3. Design and Implement Algorithms: Design and implement algorithms to solve problems efficiently using appropriate data structures, considering factors such as time complexity, space complexity, and code readability. 4. Apply Data Structures to Real-World Problems: Apply data structures to real-world scenarios, such as database management, network routing, and algorithmic problem-solving, effectively solving complex problems using the appropriate data structure and algorithmic approach. 5. Evaluate and Optimize Performance: Evaluate the performance of data structures and algorithms, identify bottlenecks, and optimize their efficiency through algorithmic improvements or selecting more suitable data structures. 6. Collaborate in Team Projects: Work effectively in teams to design and implement data structure-related projects, collaborating with team members to achieve project objectives and deliver high-quality solutions. 7. Apply Critical Thinking and Problem-Solving Skills: Apply critical thinking and problem-solving skills to analyze problems, break them down into smaller subproblems, and devise effective data structure-based solutions. 8. Communicate Data Structure Concepts: Communicate data structure concepts and solutions effectively, both orally and in written form, using appropriate terminology and visual representations. 9. Continuously Learn and Adapt: Recognize the dynamic nature of data structures and algorithms, and demonstrate the ability to learn and adapt to new data structures and algorithmic techniques as they emerge in the evolving field of computer science.
Indicative Contents المحتويات الإرشادية	The indicative contents of the "Data Structures" course may include: 1. Introduction to Data Structures: o Overview of data structures and their significance in problem-solving. o Basic terminology and concepts related to data structures. o Abstract data types and their implementation. 2. Arrays and Linked Lists: o Array representation and operations (insertion, deletion, searching). o Singly linked lists, doubly linked lists, and circular linked lists. o Linked list operations (insertion, deletion, searching, traversal). 3. Stacks and Queues:

	○ Stack data structure and its operations (push, pop, peek). ○ Queue data structure and its operations (enqueue, dequeue). ○ Applications of stacks and queues. 4. Trees and Binary Trees: ○ Tree terminology, concepts, and properties. ○ Binary tree representation and traversal algorithms (preorder, inorder, postorder). ○ Binary search trees and their operations (insertion, deletion, searching). 5. Heaps and Priority Queues: ○ Heap data structure and its properties. ○ Priority queue implementation using heaps. ○ Heap operations (insertion, deletion, heapify). 6. Graphs: ○ Graph terminology, types, and representations. ○ Graph traversal algorithms (depth-first search, breadth-first search). ○ Shortest path algorithms (Dijkstra's algorithm, Bellman-Ford algorithm). 7. Hashing and Hash Tables: ○ Hashing concepts and techniques. ○ Hash functions and collision resolution strategies. ○ Hash table implementation and operations (insertion, deletion, searching). 8. Advanced Data Structures: ○ Advanced topics such as balanced search trees (AVL trees, Red-Black trees), B-trees, and tries. ○ Advanced graph algorithms (minimum spanning trees, topological sorting). ○ Advanced hashing techniques (dynamic hashing, cuckoo hashing). 9. Applications and Case Studies: ○ Real-world applications of data structures in software development, database management, networking, and other domains. ○ Case studies highlighting the selection and utilization of appropriate data structures for specific problems. The above indicative contents provide a broad overview of the topics typically covered in a "Data Structures" course. The actual course content may vary depending on the specific curriculum and instructor.

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	The "Data Structures" course can be effectively taught using a combination of learning and teaching strategies, including: 1. Lectures: Engaging lectures delivered by the instructor to introduce and explain key concepts, theories, and techniques related to data structures.

	The lectures can include real-world examples and demonstrations to enhance understanding. 2. Hands-on Programming Exercises: Providing students with programming exercises to implement and manipulate data structures using programming languages. This allows students to gain practical experience in implementing and working with data structures. 3. Group Discussions and Peer Learning: Encouraging group discussions and peer learning activities where students can collaborate, share their knowledge, and solve problems together. This promotes active learning and helps reinforce understanding of data structure concepts. 4. Case Studies and Real-World Applications: Presenting case studies and real-world examples that demonstrate the practical applications of data structures. This helps students understand how data structures are used in various domains, such as software development, networking, and database management. 5. Problem-Solving Sessions: Conducting problem-solving sessions where students are given challenging problems to solve using appropriate data structures and algorithms. This helps develop their problem-solving skills and strengthens their understanding of data structure concepts. 6. Visualizations and Interactive Tools: Utilizing visualizations and interactive tools, such as animations and simulations, to illustrate the internal workings of data structures and algorithms. This visual approach enhances comprehension and helps students grasp complex concepts more easily. 7. Assessments and Feedback: Administering regular assessments, such as quizzes and assignments, to evaluate students' understanding and progress. Providing timely and constructive feedback on their work helps them identify areas of improvement and reinforces their learning. 8. Guest Lectures and Industry Experts: Inviting guest lecturers and industry experts to share their insights and experiences related to data structures. This provides students with a broader perspective and exposes them to real-world applications and challenges. 9. Online Resources and Self-Study: Recommending online resources, textbooks, and tutorials for self-study. This allows students to explore additional materials at their own pace and deepen their understanding of data structures. 10. Project-based Learning: Assigning projects that require students to design and implement solutions using data structures. This encourages creativity, problem-solving, and practical application of learned concepts. By employing these strategies, students can actively engage with the course material, develop a solid understanding of data structures, and acquire the necessary skills to apply them effectively in various contexts.

Delivery Plan (Weekly Syllabus)

المنهاج الاسبوعي النظري

	Material Covered
Week 1	Introduction to Data Structures and Algorithms
Week 2-3	Stacks and Queues
Week 4-5	Trees and Binary Trees
Week 6	Heaps and Priority Queues
Week 7-8	Graphs
9	Mid-Term Exam
Week 10-11	Hashing and Hash Tables
Week 12-13	Advanced Data Structures
Week 14	Advanced Graph Algorithms
Week 15	Review

Delivery Plan (Weekly Lab. Syllabus) المنهاج الاسبوعي للمختبر	
	Material Covered
Week 1	Introduction to Programming Environment 2. Setting up the programming environment (IDE, compiler, etc.) 3. Writing and executing a simple program in a programming language
Week 2	Array Manipulation 8. Implementing basic array operations (insertion, deletion, searching) 9. Analyzing the time complexity of array operations
Week 3	Linked List Implementation • Implementing a linked list data structure • Performing operations on a linked list (insertion, deletion, traversal)
Week 4	Stack and Queue Implementation 8. Implementing a stack using arrays and linked lists 9. Implementing a queue using arrays and linked lists 10. Performing stack and queue operations
Week 5	Tree Traversals 8. Implementing tree data structures (binary tree, binary search tree)

	9. Performing tree traversals (pre-order, in-order, post-order)
Week 6	Heap Operations 8. Implementing a heap data structure 9. Performing heap operations (insertion, deletion, heapify)
Week 7-8	Graph Traversals • Implementing a graph data structure (adjacency matrix, adjacency list) • Performing graph traversals (depth-first search, breadth-first search)
Week 9	Midterm Exam
Week 10-11	Hash Table Implementation • Implementing a hash table data structure • Handling collisions using separate chaining or open addressing
Week 12-13	Balanced Binary Search Tree • Implementing a balanced binary search tree (AVL tree, red-black tree) • Performing operations on the balanced binary search tree (insertion, deletion, search)
Week 14	Graph Algorithms • Implementing graph algorithms (Dijkstra's algorithm, Kruskal's algorithm) • Analyzing the time complexity of graph algorithms
Week 15	Review

Learning and Teaching Resources

مصادر التعلم والتدريس

	Text	Available in the Library?
Required Texts	Data Structure and Program Design in C++, by Robert Kruse	yes
Recommended Texts	Data Structure and Algorithm Analysis in C++, by Mark Allen Weiss	no
Websites		

MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information معلومات المادة الدراسية		
Module Title	**Object Oriented Programming**	**Module Delivery**

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Objectives أهداف المادة الدراسية	1. Advanced OOP Concepts: • Understand and apply advanced object-oriented programming concepts, such as inheritance, polymorphism, and encapsulation. • Design and implement complex class hierarchies using inheritance and composition. • Utilize advanced OOP techniques to develop modular and reusable code. 2. Design Patterns and Software Architecture: • Explore commonly used design patterns and their application in software development. • Understand architectural patterns and their role in designing scalable and maintainable software systems. • Apply design patterns and software architecture principles to solve real-world problems. 3. Data Structures and Algorithms: • Understand advanced data structures, such as trees, graphs, and hash tables, and their implementation in Python. • Analyze algorithmic complexity and choose appropriate algorithms for different problem-solving scenarios. • Implement and optimize algorithms for searching, sorting, and graph traversal. 4. Exception Handling and Error Management: • Master advanced exception handling techniques, such as exception chaining and custom exception classes. • Understand and apply defensive programming techniques to handle errors and unexpected situations. • Develop error handling strategies for robust and reliable software. 5. Concurrency and Parallel Programming: • Understand the challenges and principles of concurrent programming. • Utilize multithreading and multiprocessing to write concurrent Python programs. • Implement synchronization mechanisms and handle race conditions in concurrent code. 6. Testing and Debugging: • Learn advanced techniques for testing Python code, including unit testing, integration testing, and test-driven development (TDD). • Apply debugging strategies and tools to identify and fix software defects. • Develop a comprehensive testing and debugging mindset for producing high-quality code.

	7. GUI Development and User Experience: • Gain proficiency in developing graphical user interfaces (GUIs) using Python frameworks such as Tkinter, PyQt, or wxPython. • Understand user experience (UX) principles and design intuitive and user-friendly interfaces. • Incorporate event-driven programming to handle user interactions in GUI applications. 8. Software Development Best Practices: • Learn and apply software development best practices, including code organization, documentation, and version control. • Collaborate effectively in software development teams using version control systems like Git. • Understand the importance of code maintainability, scalability, and reusability.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	1. Advanced Understanding of OOP Concepts: • Demonstrate a deep understanding of advanced object-oriented programming concepts, including inheritance, polymorphism, and encapsulation. • Apply advanced OOP techniques to design and implement complex software systems. • Analyze and evaluate different approaches to solve programming problems using OOP principles. 2. Design Patterns and Software Architecture: • Apply various design patterns to solve software design problems effectively. • Design software architectures that are modular, maintainable, and scalable. • Analyze and evaluate different software architectural patterns for different types of applications. 3. Proficiency in Data Structures and Algorithms: • Implement and analyze advanced data structures, such as trees, graphs, and hash tables, using Python. • Design and optimize algorithms for efficient data manipulation and problem-solving. • Apply algorithmic thinking and problem-solving skills to solve complex programming challenges. 4. Robust Exception Handling and Error Management: • Implement advanced exception handling techniques to handle errors and exceptional situations in software. • Design error handling strategies to ensure robustness and reliability of software systems. • Analyze and debug complex software issues related to error management and exception handling. 5. Concurrent and Parallel Programming: • Demonstrate an understanding of concurrent programming principles and techniques. • Utilize multithreading and multiprocessing in Python to write concurrent and parallel programs. • Implement synchronization mechanisms and handle concurrency-related issues.

	6. Effective Testing and Debugging: • Apply advanced testing techniques, such as unit testing, integration testing, and test-driven development (TDD), to ensure software quality. • Use debugging tools and strategies to identify and resolve complex software defects. • Develop a systematic approach to testing and debugging software systems. 7. Advanced GUI Development and User Experience: • Design and develop sophisticated graphical user interfaces (GUIs) using Python frameworks such as Tkinter, PyQt, or wxPython. • Apply user experience (UX) principles to create intuitive and user-friendly interfaces. • Implement event-driven programming to handle user interactions and enhance user experience. 8. Application of Software Development Best Practices: • Apply software development best practices, including code organization, documentation, and version control, to develop high-quality software. • Collaborate effectively in software development teams, demonstrating good teamwork and communication skills. • Demonstrate an understanding of the importance of code maintainability, scalability, and reusability. 9. Real-World Application Development: • Design, implement, and test larger-scale software projects using object-oriented programming principles. • Apply software engineering principles and techniques to manage project scope, requirements, and timelines. • Demonstrate proficiency in project planning, teamwork, and project management.
Indicative Contents المحتويات الإرشادية	1. Advanced OOP Concepts: • Inheritance: Advanced inheritance concepts such as multiple inheritance, method resolution order (MRO), and mixins. • Polymorphism: Advanced polymorphism techniques including method overriding, abstract base classes (ABCs), and interfaces. • Composition: Utilizing composition over inheritance to design and implement complex class relationships. 2. Design Patterns and Software Architecture: • Introduction to design patterns: Understanding and applying common design patterns such as Singleton, Factory, Observer, and Strategy. • Software architecture principles: Exploring architectural patterns like Model-View-Controller (MVC) and understanding their application in software development. • Component-based architecture: Designing and implementing software using component-based architectural patterns. 3. Data Structures and Algorithms: • Advanced data structures: Implementation and application of advanced data structures like balanced search trees, heaps, and graphs. • Algorithm analysis: Analyzing the time and space complexity of algorithms and choosing the appropriate algorithmic solutions for different problem domains. • Sorting and searching algorithms: Implementing and analyzing various

	sorting and searching algorithms, including quicksort, mergesort, binary search, and more. 4. Exception Handling and Error Management: • Advanced exception handling: Handling and propagating exceptions, exception chaining, and creating custom exception classes. • Defensive programming: Implementing defensive programming techniques to handle errors and edge cases in software. • Error management strategies: Designing error handling strategies to ensure fault tolerance and reliability in software systems. 5. Concurrency and Parallel Programming: • Introduction to concurrency: Understanding the challenges and principles of concurrent programming. • Threading and multiprocessing: Implementing multithreading and multiprocessing techniques in Python for concurrent and parallel programming. • Synchronization and coordination: Utilizing synchronization mechanisms like locks, semaphores, and condition variables to handle shared resources and coordinate concurrent tasks. 6. Testing and Debugging: • Advanced testing techniques: Implementing unit tests, integration tests, and test-driven development (TDD) approaches for robust software testing. • Debugging strategies: Applying advanced debugging techniques and tools to identify and fix software defects. • Test coverage and code quality: Understanding the importance of code coverage and maintaining high code quality through testing and debugging. 7. GUI Development and User Experience: • GUI frameworks: Exploring advanced GUI frameworks in Python, such as PyQt, wxPython, or Kivy, for developing interactive graphical user interfaces. • User experience design: Incorporating user-centered design principles to create intuitive and visually appealing user interfaces. • Event-driven programming: Utilizing event-driven programming to handle user interactions and create responsive GUI applications. 8. Software Development Best Practices: • Code organization and modularity: Applying modular design principles and organizing code into reusable and maintainable components. • Documentation and commenting: Writing clear and comprehensive documentation and comments to enhance code readability and understandability. • Version control: Utilizing version control systems, such as Git, for collaborative software development and code management. 9. Real-World Application Development: • Large-scale project development: Working on larger-scale projects that involve designing, implementing, and testing complex software systems. • Project planning and management: Understanding project management methodologies and applying them to effectively plan and manage software development projects. • Team collaboration: Collaborating with peers in a team environment,

	demonstrating effective communication and teamwork skills.

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	1. Active Learning and Problem-Solving Approach: • Emphasize active learning strategies, such as hands-on coding exercises, group discussions, and problem-solving activities. • Encourage students to apply their knowledge of OOP principles and Python programming to real-world scenarios and projects. • Provide opportunities for students to actively engage with the material through coding challenges, case studies, and practical assignments. 2. Project-Based Learning: • Implement a project-based approach where students work on larger-scale programming projects that require the application of advanced OOP concepts. • Assign projects that involve designing, implementing, and testing software systems using Python and OOP principles. • Encourage students to collaborate in teams, simulate real-world development environments, and manage project requirements and deadlines. 3. Practical Coding Exercises and Assignments: • Provide a variety of coding exercises and assignments that focus on advanced OOP topics, design patterns, algorithms, and software architecture. • Include programming assignments that require students to implement complex OOP concepts, solve algorithmic problems, and design efficient data structures. • Provide feedback and guidance on students' code to promote good programming practices and enhance their understanding of OOP in Python. 4. Code Reviews and Peer Collaboration: • Incorporate code review sessions where students review and provide feedback on each other's code, promoting code quality and best practices. • Encourage peer collaboration and teamwork, fostering communication and problem-solving skills in a professional software development context. • Utilize version control systems (e.g., Git) to facilitate code sharing, collaboration, and tracking of project development. 5. Integration of Design Patterns and Software Architecture: • Introduce design patterns and software architecture principles in a practical context, demonstrating their relevance and benefits in software development. • Guide students to identify and apply appropriate design patterns and architectural patterns in their projects.

	• Discuss case studies and examples of real-world applications where design patterns and software architecture have been effectively used. 6. Practical Testing and Debugging Techniques: • Teach advanced testing techniques, such as unit testing, integration testing, and test-driven development (TDD), to ensure software quality and reliability. • Emphasize the importance of systematic debugging approaches and the use of debugging tools to identify and resolve software defects. • Provide practical examples and exercises that require students to write comprehensive test cases and debug complex code scenarios. 7. Practical Application of GUI Development: • Provide practical assignments and projects that require students to develop GUI applications using Python frameworks like Tkinter, PyQt, or wxPython. • Incorporate user experience (UX) principles and usability testing to enhance the design and functionality of GUI applications. • Encourage students to incorporate event-driven programming concepts to handle user interactions and create interactive interfaces. 8. Exposure to Real-World Software Development Practices: • Introduce students to software development best practices, including code organization, documentation, and version control. • Familiarize students with collaborative software development tools and techniques, such as code repositories and issue tracking systems. • Discuss industry trends, emerging technologies, and the importance of continuous learning in the field of object-oriented programming.

Delivery Plan (Weekly Syllabus)

المنهاج الاسبوعي النظري

	Material Covered
Week 1	Introduction to Object Oriented Programing and Structural Programming
Week 2-3	Introduction to Classes and Objects
Week 4	Encapsulation and Access modifiers
Week 5	Abstraction
Week 6-7	Inheritance
Week 8-9	Polymorphism
Week 10	Mid Term Examination
Week 11	Operator Overloading
Week 12	Operator Overriding
Week 13-15	Project and Presentation

Delivery Plan (Weekly Lab. Syllabus)

المنهاج الاسبوعي للمختبر

	Material Covered
Week 1	Week 1: Introduction to Object-Oriented Programming and Structural Programming • Overview of the principles and concepts of object-oriented programming (OOP) • Introduction to the fundamentals of structural programming • Discussion on the advantages and characteristics of OOP
Week 2-3	Week 2 - 3: Introduction to Classes and Objects • Understanding the concept of classes and objects in OOP • Creating and defining classes in C++ • Exploring object instantiation and member functions
Week 4	Week 4: Encapsulation and Access Modifiers • Understanding encapsulation and its importance in OOP • Exploring access modifiers (public, private, protected) • Discussion on data hiding and encapsulation principles
Week 5	Week 5: Abstraction • Introduction to abstraction in OOP • Understanding abstract classes and interfaces • Implementing abstraction in C++ using pure virtual functions
Week 6-7	Week 6-7: Inheritance • Exploring the concept of inheritance in OOP • Implementing inheritance in C++ through derived classes • Discussing the different types of inheritance (single, multiple, multilevel, hierarchical)
Week 8-9	Week 8-9: Polymorphism • Understanding polymorphism and its significance in OOP • Exploring function overloading and overriding • Implementing polymorphism through virtual functions in C++
Week 10	Week 10: Midterm Examination • Midterm examination covering topics from weeks 1-9 • Review of previous topics and discussion of any questions or concerns

Week 11	Week 11: Operator Overloading • Introduction to operator overloading in C++ • Overloading unary and binary operators • Exploring the use of friend functions for operator overloading
Week 12	Week 12: Operator Overriding • Understanding the concept of operator overriding • Overriding base class operators in derived classes • Discussion on the limitations and best practices of operator overriding
Week 13-15	Week 13-15: Project and Presentation • Working on a project that incorporates the principles and concepts covered in the course • Planning, designing, and implementing an object-oriented program • Preparing a presentation to showcase the project and its features

Learning and Teaching Resources
مصادر التعلم والتدريس

	Text	Available in the Library?
Required Texts	C++ Primer (5th Edition) 5th Edition , by Stanley Lippman (Author), Josée Lajoie (Author), Barbara Moo (Author)	
Recommended Texts		
Websites		

MODULE DESCRIPTION FORM
نموذج وصف المادة الدراسية

Module Information معلومات المادة الدراسية		
Module Title	Website Design &Programming I	**Module Delivery**

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Objectives أهداف المادة الدراسية	1. Understanding Web Technologies: To provide students with a foundational understanding of web technologies, including HTML, CSS, and JavaScript, and their role in building websites and web applications. 2. Website Structure and Design: To teach students how to create well-structured and visually appealing websites using HTML and CSS, focusing on concepts such as layout, typography, color schemes, and responsive design. 3. Interactive Web Elements: To enable students to incorporate interactive elements into websites using JavaScript, such as form validation, event handling, and dynamic content manipulation. 4. Client-Server Communication: To introduce students to the basics of client-server communication in web development, including sending and receiving data from a server using HTTP requests and APIs. 5. Web Development Tools: To familiarize students with popular web development tools, such as text editors, version control systems, and debugging tools, and teach them how to use these tools effectively in their development workflow. 6. Web Accessibility: To emphasize the importance of web accessibility and teach students how to design and develop websites that are inclusive and accessible to users with disabilities. 7. Project Development: To provide students with hands-on experience in developing a complete web project, from conceptualization to deployment, while applying the concepts and techniques learned throughout the course. 8. Collaboration and Communication: To promote teamwork and effective communication skills by encouraging students to work collaboratively

	on group projects, communicate project requirements and progress, and provide constructive feedback to their peers. 9. Problem-Solving and Debugging: To enhance students' problem-solving skills and teach them how to debug and troubleshoot common issues in web development, fostering a systematic and analytical approach to solving technical problems. 10. Professionalism and Ethical Considerations: To instill professional ethics and good practices in web development, including copyright and intellectual property considerations, respecting user privacy, and adhering to industry standards and best practices.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	• Knowledge of Web Technologies: Gain a solid understanding of web technologies, including HTML, CSS, and JavaScript, and their role in web development. • Website Creation: Design and develop well-structured websites using HTML and CSS, considering factors such as layout, typography, color schemes, and responsive design. • Interactive Elements: Implement interactive features on websites using JavaScript, such as form validation, event handling, and dynamic content manipulation. • Client-Server Communication: Understand the basics of client-server communication in web development, including making HTTP requests and working with APIs to retrieve and send data. • Use of Web Development Tools: Utilize popular web development tools, such as text editors, version control systems, and debugging tools, to enhance productivity and efficiency in web development projects. • Web Accessibility: Apply principles of web accessibility to ensure websites are inclusive and accessible to users with disabilities, following accessibility guidelines and best practices. • Project Development: Develop a complete web project, applying the knowledge and skills acquired throughout the course, from planning and design to implementation and deployment. • Collaboration and Communication: Collaborate effectively with team members, communicate project requirements and progress, and provide constructive feedback to peers in group projects. • Problem-Solving and Debugging: Demonstrate problem-solving skills and the ability to debug and troubleshoot issues in web development projects, using

	a systematic and analytical approach. • Professionalism and Ethical Considerations: Understand and adhere to professional ethics and considerations in web development, including respecting copyright and intellectual property, protecting user privacy, and following industry standards and best practices
Indicative Contents المحتويات الإرشادية	The indicative contents for Web Development 1 may include: 1. Introduction to Web Technologies: • Overview of web development concepts and technologies • Understanding the client-server architecture and how the web works 2. HTML Fundamentals: • HTML syntax and structure • Working with tags, attributes, and elements • Creating hyperlinks, lists, tables, and forms 3. CSS Basics: • Introduction to Cascading Style Sheets (CSS) • Applying styles to HTML elements • Working with selectors, properties, and values • Managing layout, typography, and colors 4. JavaScript Fundamentals: • Introduction to JavaScript programming language • Variables, data types, and operators • Conditional statements and loops • Functions and event handling 5. Responsive Web Design: • Designing websites that adapt to different screen sizes and devices • Using media queries and viewport meta tags • Implementing responsive layouts and navigation menus 6. Web Accessibility: • Understanding the importance of web accessibility • Applying accessibility principles and techniques • Testing and optimizing websites for accessibility 7. Introduction to Server-side Technologies: • Overview of server-side programming languages and frameworks

	• Introduction to databases and server-side scripting • Basics of server-side development and interacting with databases 8. Introduction to Version Control: • Understanding the concept of version control and its importance in web development • Using Git for version control and collaboration • Branching, merging, and resolving conflicts 9. Web Project Development: • Planning and organizing a web development project • Creating wireframes and mockups • Implementing the project using HTML, CSS, and JavaScript 10. Deployment and Maintenance: • Uploading and hosting a website on a server • Performing maintenance tasks and updates • Testing and troubleshooting common issues Please note that the above contents are indicative and may vary depending on the specific curriculum and institution offering the course

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	1. Lectures: In-class lectures can be used to introduce and explain key concepts, programming languages, and techniques related to web development. The instructor can provide examples and demonstrations to illustrate the concepts. 2. Hands-on Practice: Hands-on practice is essential for web development. Students can engage in practical exercises and coding activities during the class or in dedicated lab sessions. This allows them to apply the knowledge gained and practice coding HTML, CSS, and JavaScript. 3. Project-based Learning: Assigning small projects or tasks related to web development can provide students with real-world scenarios to apply their skills and knowledge. Working on projects helps them develop problem-solving abilities, collaboration skills, and practical experience in building websites. 4. Group Discussions and Peer Learning: Encouraging group discussions and peer learning can enhance understanding and knowledge retention. Students can discuss challenges, exchange ideas, and collaborate on problem-solving. This fosters a collaborative learning environment and allows students to learn from each other's experiences.

	5. Online Resources and Tutorials: Providing students with online resources, tutorials, and documentation can supplement classroom learning. These resources can include video tutorials, coding exercises, interactive websites, and documentation of programming languages and frameworks. 6. Code Reviews and Feedback: Conducting code reviews and providing feedback on student projects or assignments can help improve their coding skills. Feedback can be provided by the instructor or through peer code reviews. This helps students understand best practices, identify areas for improvement, and learn from their mistakes. 7. Guest Speakers and Industry Insights: Inviting guest speakers from the industry or web development professionals can provide valuable insights and real-world experiences to students. They can share their expertise, industry trends, and challenges in web development, inspiring students and bridging the gap between academia and industry. 8. Assessment and Evaluation: Assessments can include quizzes, assignments, projects, and exams to evaluate students' understanding and progress. This allows the instructor to gauge their knowledge and provide constructive feedback for improvement. 9. Continuous Learning and Updates: Web development is a rapidly evolving field. It is important to emphasize the need for continuous learning and staying updated with the latest technologies, frameworks, and best practices. Encouraging students to explore online resources, attend workshops, and engage in self-directed learning can help them keep up with industry trends. It is worth noting that the selection and implementation of these strategies may vary based on the specific educational institution, class size, resources available, and the preferences of the instructor.

Delivery Plan (Weekly Syllabus)

المنهاج الاسبوعي النظري

	Material Covered
Week 1	Introduction to Web Development • Introduction to HTML and its structure • Creating a basic HTML webpage • Understanding CSS and styling web pages
Week 2-3	HTML and CSS Fundamentals • Working with text, links, images, and lists in HTML • Applying CSS styles to HTML elements

	• Introduction to responsive design principles
Week 4-5	Web Design Principles • Understanding color theory and typography in web design • Creating layouts using CSS positioning and flexbox • Introduction to CSS frameworks like Bootstrap
Week 6-7	Multimedia and Forms • Adding images, videos, and audio to web pages • Creating forms and handling user input • Validating form data using HTML5 and JavaScript
Week 8-9	: Introduction to JavaScript • Fundamentals of JavaScript programming • Working with variables, data types, and operators • Writing JavaScript functions and control structures
Week 10-11	JavaScript DOM Manipulation • Accessing and modifying HTML elements using JavaScript • Handling events and creating interactive web pages • Introduction to JavaScript libraries (e.g., jQuery) •
Week 12-13	: Introduction to Front-End Frameworks • Exploring popular front-end frameworks like React or Vue.js • Building dynamic and interactive web pages using frameworks • Understanding component-based development
Week 14-15	Responsive Design and Deployment • Designing responsive websites for different devices and screen sizes • Optimizing web pages for performance • Deploying a website to a web server •

Delivery Plan (Weekly Lab. Syllabus)

المنهاج الاسبوعي للمختبر

	Material Covered
Week 1	: Introduction to HTML • Creating a basic HTML page • Adding headings, paragraphs, and lists • Working with links and images
Week 2	CSS Styling • Applying CSS styles to HTML elements • Creating and styling navigation menus • Implementing CSS layouts and positioning
Week 3	Forms and Input Validation • Creating HTML forms with various input types • Implementing client-side form validation using JavaScript • Handling form submission and processing user input
Week 4	Responsive Web Design • Designing a responsive web layout using CSS media queries • Adapting the website for different screen sizes and devices • Testing and optimizing the website for mobile devices
Week 5	Introduction to JavaScript 10. Writing basic JavaScript code 11. Manipulating the DOM using JavaScript 12. Implementing simple interactive features on a web page
Week 6	: JavaScript Functions and Events • Defining and calling JavaScript functions • Handling different types of events (e.g., click, mouseover) • Implementing event handlers and callback functions
Week 7	Introduction to Bootstrap Framework • Using Bootstrap CSS classes and components for rapid web development • Building responsive and mobile-friendly web layouts with Bootstrap • Customizing and extending Bootstrap components
Week 8	JavaScript Libraries and Frameworks • Introduction to popular JavaScript libraries (e.g., jQuery) • Exploring the features and functionalities of JavaScript frameworks (e.g., Angular,

	React) • Building interactive web applications using libraries and frameworks
Week 9	Working with APIs and JSON • Making AJAX requests to retrieve data from external APIs • Parsing and manipulating JSON data • Displaying API data on a web page
Week 10	: Introduction to Server-Side Development • Setting up a local development environment (e.g., Apache, PHP) • Writing server-side scripts using PHP • Implementing basic server-side functionality (e.g., form handling, database connection)
Week 11-14	11-14: Web Project Development • Applying the concepts learned to develop a complete web project • Planning, designing, and implementing a website or web application • Testing, debugging, and optimizing the web project
Week 15	Project Presentation and Evaluation

Learning and Teaching Resources

مصادر التعلم والتدريس

	Text	Available in the Library?
Required Texts		No
Recommended Texts		
Websites		

MODULE DESCRIPTION FORM
نموذج وصف المادة الدراسية

Module Information معلومات المادة الدراسية		
Module Title	**Network Protocol -1-**	**Module Delivery**

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Objectives أهداف المادة الدراسية	In this course, Networks department aims to achieve the following objectives: 1. Understand the fundamental concepts and principles of the TCP/IP protocol suite. 2. Explain the layered structure of the TCP/IP model and the functions of each layer. 3. Identify and describe the key protocols and components of the TCP/IP architecture. 4. Demonstrate knowledge of network infrastructures, including different topologies and connecting devices. 5. Perform IP address calculations using classful and classless addressing techniques. 6. Apply subnetting and supernetting methods to efficiently allocate IP addresses. 7. Understand the process of data delivery and routing in TCP/IP networks. 8. Configure and troubleshoot basic network connectivity using TCP/IP protocols. 9. Analyze network connectivity issues and apply appropriate solutions. 10. Apply critical thinking and problem-solving skills to address real-world networking scenarios.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	Upon successful completion of this course for the Networks department, students should be able to demonstrate the following learning outcomes: 1. Comprehensive Understanding: • Demonstrate a comprehensive understanding of the TCP/IP

	protocol suite and its components. • Explain the functions and responsibilities of each layer in the TCP/IP model. 2. IP Addressing and Subnetting: • Apply classful and classless addressing techniques for IP address allocation. • Perform subnetting and supernetting calculations to efficiently manage network segments. 3. Network Infrastructure and Connectivity: • Identify network topologies and understand the role of connecting devices. • Configure and troubleshoot basic network connectivity using TCP/IP protocols. 4. Data Delivery and Routing: • Explain the process of data delivery, including packet encapsulation, routing, and forwarding. • Analyze and troubleshoot network connectivity issues using routing protocols. 5. Critical Thinking and Problem-Solving: • Apply critical thinking skills to solve real-world networking scenarios. • Evaluate and select appropriate IP addressing strategies based on network requirements. 6. Communication and Collaboration: • Communicate effectively about TCP/IP concepts and network-related issues. • Collaborate with peers to solve problems and share knowledge. 7. Practical Application: • Apply acquired knowledge to design, configure, and troubleshoot TCP/IP networks. • Implement effective IP addressing schemes and network segmentation strategies. 8. Lifelong Learning: • Recognize the importance of continuous learning in the field of networking. • Stay updated with new protocols and technologies related to TCP/IP networking.
Indicative Contents المحتويات الإرشادية	The indicative contents of this course for the computer department may include the following topics: 1. Internet Introduction • History and evolution of the Internet • Key milestones and developments

	• Internet architecture and key protocols • Internet governance and organizations 2. TCP/IP Protocol Suite Layers • Introduction to the TCP/IP protocol suite • Overview of the TCP/IP layers (Application, Transport, Internet, Link) • Functions and responsibilities of each layer • Protocols and services associated with each layer 3. Infrastructure Network and Connecting Devices • Overview of network infrastructures • Network topologies: bus, star, ring, mesh, etc. • Common connecting devices: switches, routers, hubs • Benefits and limitations of different network infrastructures 4. Classful Addressing and Special Addressing • Introduction to classful addressing • IP address classes and ranges (Class A, B, C, D, E) • Special IP addresses: loopback, broadcast, multicast • Address exhaustion issues with classful addressing 5. Subnetting and Supernetting • Introduction to subnetting and its need • Subnet masks and subnet addressing • Calculating subnet addresses and broadcast addresses • Supernetting and route aggregation for efficient addressing 6. Classless Addressing • Introduction to classless addressing • Classless Inter-Domain Routing (CIDR) • Variable Length Subnet Masking (VLSM) • Address allocation and route summarization in classless addressing 7. Delivery and Routing • Data encapsulation and decapsulation process • Overview of packet delivery and routing • Static routing and dynamic routing protocols (RIP, OSPF) • Routing table configuration and routing decision process

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	Learning and teaching strategies for this course for the Network department can include a combination of the following: 46. Lectures: Engage students through informative lectures that cover

	theoretical concepts and provide an overview of key topics. Use multimedia resources, visuals, and real-world examples to enhance understanding. 47. Group Discussions and Collaborative Learning: Encourage group discussions and collaborative activities to foster interaction and knowledge sharing among students. Assign group projects or case studies that require teamwork and problem-solving. 48. Online Resources and Multimedia: Utilize online resources, interactive tutorials, and multimedia materials to supplement learning. This can include video lectures, online quizzes, virtual labs, and interactive modules. 49. Assignments and Projects: Assign individual and group projects that require students to apply their knowledge and skills to solve real-world problems or complete practical tasks. This promotes critical thinking, problem-solving, and practical application of concepts. 50. Assessments and Feedback: Conduct regular assessments, quizzes, and examinations to evaluate students' understanding of the course material. Provide timely and constructive feedback to help students identify areas of improvement. 51. Industry Visits and Field Trips: Organize visits to IT companies, data centers, or relevant organizations to expose students to real-world IT environments. This provides valuable industry insights and networking opportunities. 52. Online Discussion Forums and Communication Platforms: Establish online discussion forums or communication platforms where students can ask questions, share resources, and engage in discussions outside of the classroom. These strategies promote active learning, practical application of knowledge, and engagement with the subject matter. They cater to different learning styles and encourage students to develop critical thinking, problem-solving, and communication skills necessary for success in this field.

Delivery Plan (Weekly Syllabus)

المنهاج الاسبوعي النظري

	Material Covered
Week 1	• Introduction to TCP/IP and the Internet
Week 2	• TCP/IP Protocol Suite Layers
Week 3	• Infrastructure Network and Connecting Devices
Week 4-6	• Classful Addressing and Special Addressing

Week 7-9	• Subnetting and Supernetting
Week 10	• Mid Term
Week 11-12	• Classless Addressing
Week 13-14	• Delivery and Routing
Week 15	• Review
Week 16	**Preparatory week before the final Exam**

Learning and Teaching Resources مصادر التعلم والتدريس		
	Text	**Available in the Library?**
Required Texts	TCP/IP PROTOCOL SUITE	
Recommended Texts		
Websites		

MODULE DESCRIPTION FORM
نموذج وصف المادة الدراسية

Module Information معلومات المادة الدراسية		
Module Title	**Microprocessors & Interfacing**	**Module Delivery**

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Objectives أهداف المادة الدراسية	The objectives of the course "Microprocessors and Interfacing" are: 1. Understand microprocessor architecture: Gain a deep understanding of the internal structure, components, and functioning of microprocessors, particularly the 8086 family. 2. Learn assembly language programming: Develop proficiency in programming microprocessors using assembly language, including data manipulation, arithmetic operations, and control flow. 3. Explore interfacing techniques: Acquire knowledge of various interfacing techniques to connect microprocessors with peripheral devices, such as memory, I/O ports, and interrupt controllers. 4. Develop troubleshooting skills: Learn to identify and resolve common issues in microprocessor-based systems, including interfacing problems and software errors. 5. Apply memory organization concepts: Understand memory organization principles, including addressing modes and memory hierarchy, to efficiently manage data in microprocessor systems. 6. Enhance practical skills: Gain hands-on experience through laboratory experiments, programming microprocessors, designing interfaces, and implementing projects. 7. Foster teamwork and communication: Collaborate effectively in team-based projects, developing communication and teamwork skills essential for working in the field of microprocessors. 8. Prepare for advanced studies: Develop a strong foundation in microprocessor architecture and interfacing, providing a solid base for further studies or specialization in related areas. 9. Foster critical thinking: Analyze and evaluate microprocessor systems and interfaces, identifying areas for improvement and optimization to enhance performance. 10. Understand real-world applications: Gain an understanding of the applications and impact of microprocessors in various industries,

	including embedded systems, robotics, and automation. By achieving these objectives, students will be well-equipped with the necessary knowledge and skills to design, program, and interface microprocessors effectively in a range of applications.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	Upon successful completion of the course "Microprocessors and Interfacing," students will be able to: 1. Demonstrate a comprehensive understanding of microprocessor architecture, including the internal structure, registers, memory organization, and addressing modes. 2. Write efficient assembly language programs for microprocessors, demonstrating proficiency in data manipulation, arithmetic operations, and control flow. 3. Design and implement interfaces between microprocessors and peripheral devices, including memory, I/O ports, and interrupt controllers. 4. Troubleshoot and debug common issues in microprocessor-based systems, employing problem-solving skills and analytical thinking. 5. Apply memory organization concepts to effectively manage data in microprocessor systems, including addressing modes and memory hierarchy. 6. Work effectively in a team, collaborating with peers to design and implement microprocessor-based projects. 7. Communicate technical concepts and findings clearly and concisely, both orally and in written reports. 8. Critically analyze and evaluate microprocessor systems and interfaces, identifying areas for improvement and optimization. 9. Demonstrate knowledge of real-world applications of microprocessors, understanding their impact in various industries. 10. Adapt and apply their knowledge and skills to future technological advancements in microprocessor systems and interfacing. These learning outcomes reflect the students' ability to understand, apply, and analyze the concepts and techniques related to microprocessors and interfacing, preparing them for further studies or professional careers in the field.
Indicative Contents المحتويات الإرشادية	The indicative contents of the course "Microprocessors and Interfacing" may include: 1. Introduction to Microprocessors: o Basics of microprocessors and their applications o Evolution of microprocessor architecture o Overview of the 8086 family of microprocessors 2. Microprocessor Architecture and Organization: o Internal architecture of microprocessors o Registers and their functions o Memory organization and addressing modes 3. Assembly Language Programming: o Introduction to assembly language programming o Instruction set architecture of the 8086 microprocessor o Data manipulation, arithmetic operations, and control flow in

	assembly language 4. I/O Interfacing: o Input/output interfacing techniques o Interfacing with memory, I/O ports, and interrupt controllers o Handling interrupts and interrupt service routines 5. Memory Organization: o Memory hierarchy and cache memory o Addressing modes and memory segmentation o Memory mapping and address decoding techniques 6. Interrupts and Interrupt Handling: o Types of interrupts and their sources o Interrupt request (IRQ) handling and priority schemes o Writing interrupt service routines 7. Practical Lab Sessions: o Hands-on experiments to reinforce concepts learned in lectures o Programming microprocessors using assembly language o Designing and implementing interfaces with peripheral devices 8. Troubleshooting and Debugging: o Techniques for identifying and resolving common issues in microprocessor systems o Debugging tools and methodologies for diagnosing problems 9. Advanced Topics: o Introduction to advanced microprocessor architectures and their features o Overview of parallel processing and multi-core systems o Introduction to embedded systems and real-time operating systems 10. Applications and Case Studies: o Real-world applications of microprocessors in various industries o Case studies highlighting the use of microprocessors in different systems These indicative contents provide a general overview of the topics covered in the course, ensuring students gain a comprehensive understanding of microprocessors, assembly language programming, interfacing techniques, and practical application of concepts in laboratory sessions.

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	The course "Microprocessors and Interfacing" can be taught using a variety of effective learning and teaching strategies, including: 1. Lectures: Engaging lectures delivered by the instructor to introduce and explain key concepts, theories, and principles related to microprocessors and interfacing. 2. Hands-on Lab Sessions: Practical lab sessions where students can apply their knowledge and skills by working with microprocessors, programming in assembly language, and designing and implementing interfaces with peripheral devices.

	3. Interactive Discussions: Facilitating interactive discussions among students to encourage active participation and exchange of ideas, allowing them to deepen their understanding of the course material. 4. Case Studies: Presenting real-world case studies and examples that demonstrate the practical applications of microprocessors and interfacing techniques in various industries, such as robotics, automation, and embedded systems. 5. Group Projects: Assigning group projects that require students to collaborate and apply their knowledge to design and develop microprocessor-based systems or applications. 6. Problem-Solving Exercises: Providing problem-solving exercises and assignments that challenge students to apply their understanding of microprocessors and interfacing to solve complex problems. 7. Guest Lectures and Industry Visits: Inviting guest speakers from the industry or organizing visits to relevant companies or organizations to expose students to real-world practices and the latest trends in microprocessors and interfacing. 8. Online Resources: Utilizing online resources such as interactive simulations, tutorials, and supplementary materials to enhance understanding and provide additional learning opportunities outside of the classroom. 9. Assessment Methods: Employing a combination of quizzes, assignments, lab reports, and examinations to assess students' understanding and mastery of the course material. 10. Feedback and Reflection: Providing regular feedback to students on their progress and performance, as well as encouraging self-reflection to promote continuous improvement and learning. These learning and teaching strategies aim to create an engaging and interactive learning environment that fosters critical thinking, problem-solving skills, and practical application of microprocessors and interfacing concepts.

Delivery Plan (Weekly Syllabus)

المنهاج الاسبوعي النظري

	Material Covered
Week 1	Introduction to Microprocessors • Overview of microprocessors and their applications • Evolution of microprocessor architecture • Introduction to the 8086 family of microprocessors
Week 2-4	Microprocessor Architecture and Organization • Internal architecture of microprocessors • Registers and their functions • Memory organization and addressing modes

Week 5-7	Assembly Language Programming • Introduction to assembly language programming • Instruction set architecture of the 8086 microprocessor • Data manipulation, arithmetic operations, and control flow in assembly language
Week 8-10	I/O Interfacing • Input/output interfacing techniques • Interfacing with memory, I/O ports, and interrupt controllers • Handling interrupts and writing interrupt service routines
Week 11-12	Memory Organization • Memory hierarchy and cache memory • Addressing modes and memory segmentation • Memory mapping and address decoding techniques
Week 13-14	Advanced Topics • Introduction to advanced microprocessor architectures and features • Overview of parallel processing and multi-core systems • Introduction to embedded systems and real-time operating systems
Week 15	Review

Delivery Plan (Weekly Lab. Syllabus) المنهاج الاسبوعي للمختبر	
	Material Covered
Week 1	Introduction to Microprocessors and Assembly Language Programming • Setting up the development environment (assembler and simulator) • Writing and executing simple assembly language programs to perform basic arithmetic operations
Week 2	I/O Interfacing 10. Interfacing a switch and an LED with the microprocessor 11. Writing assembly language programs to read the switch state and control the LED
Week 3	Memory Interfacing • Interfacing a 7-segment display with the microprocessor • Writing assembly language programs to display numbers and characters on the 7-segment display

Week 4	Timer and Counter Applications 11. Using timer and counter modules of the microprocessor 12. Implementing time-based applications such as blinking LEDs or generating sound tones
Week 5	Serial Communication 13. Interfacing the microprocessor with a serial communication module 14. Writing assembly language programs to send and receive data through the serial port
Week 6-7	Interrupt Handling 10. Implementing interrupt-driven I/O using the microprocessor's interrupt capabilities 11. Writing assembly language programs to handle interrupts from external devices
Week 7	Graph Traversals • Implementing a graph data structure (adjacency matrix, adjacency list) • Performing graph traversals (depth-first search, breadth-first search)
Week 8-9	Analog-to-Digital Conversion • Interfacing an ADC (Analog-to-Digital Converter) with the microprocessor • Reading analog signals and converting them to digital values using the ADC •
Week 10-11	Digital-to-Analog Conversion • Interfacing a DAC (Digital-to-Analog Converter) with the microprocessor • Generating analog signals from digital values using the DAC
Week 12-13	External Memory Interfacing • Interfacing external memory modules (e.g., RAM, ROM) with the microprocessor • Reading from and writing to external memory locations
Week 14-15	Final Project • Designing and implementing a more complex project that involves multiple peripherals and real-time operations • Integrating all the concepts learned throughout the course into a complete microprocessor-based system

Learning and Teaching Resources مصادر التعلم والتدريس		
	Text	**Available in the Library?**

Required Texts		
Recommended Texts		
Websites		

MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information معلومات المادة الدراسية		
Module Title	**Software Engineering**	**Module Delivery**
Module Type	**Core**	☒ **Theory** ☒ **Lecture** ☒ **Lab** ☐ **Tutorial** ☐ **Practical** ☐ **Seminar**

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Objectives أهداف المادة الدراسية	11. Introduction to Software Engineering: Students will gain an understanding of the fundamental concepts and principles of software engineering. They will explore the importance of software engineering in developing software systems. 12. Software Development Life Cycle: Students will learn about the different phases of the software development life cycle (SDLC), including requirements gathering, analysis, design, implementation, testing, deployment, and

	maintenance. 13. Requirements Engineering: Students will study the process of gathering, analyzing, and documenting software requirements. They will learn techniques for eliciting requirements, creating use cases, and developing requirement specifications. 14. Software Design: Students will explore software design principles and methodologies. They will learn to create high-level and low-level designs using various architectural styles and design patterns. 15. Software Implementation: Students will learn about programming practices, coding standards, and software development tools. They will gain hands-on experience in implementing software solutions using programming languages and development frameworks. 16. Software Testing and Quality Assurance: Students will understand the importance of software testing in ensuring software quality. They will learn different testing techniques, such as unit testing, integration testing, system testing, and acceptance testing.. 17. Software Maintenance and Evolution: Students will study the challenges and techniques involved in software maintenance and evolution. 18. Software Project Management: Students will be introduced to project management principles and practices specific to software development projects. They will learn about project planning, scheduling, estimation, risk management, and team collaboration. 19. Software Configuration Management: Students will understand the importance of configuration management in software development. They will learn about version control systems, change management, and release management. 20. Software Engineering Ethics and Professionalism: Students will explore ethical issues and responsibilities in software engineering. They will learn about professional codes of conduct, intellectual property rights, and privacy concerns.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	Throughout the course, students will engage in practical assignments, group projects, and case studies to apply the software engineering concepts and methodologies learned in real-world scenarios. The course prepares students for careers in software development, software project management, and related fields, providing them with the knowledge and skills to contribute effectively to software development teams and deliver high-quality software products.
Indicative Contents	The following are indicative contents that may be covered in the course: 1- Introduction to Software Engineering

المحتويات الإرشادية	2- Software Requirements Engineering 3- Software Design and Architecture 4- Software Testing and Quality Assurance 5- Software Maintenance and Evolution 6- Software Verification and Validation 7- Software Engineering Tools and Environments 8- Software Documentation and Communication 9- Software Ethics and Professionalism

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	Learning and teaching strategies for the IT basics course for the Network department can include a combination of the following: 53. Lectures: Engage students through informative lectures that cover theoretical concepts and provide an overview of key topics. Use multimedia resources, visuals, and real-world examples to enhance understanding. 54. Hands-on Labs: Provide practical lab sessions where students can apply their knowledge and skills acquired in lectures. These labs can involve hardware assembly, software installation, network configuration, programming exercises, and troubleshooting. 55. Group Discussions and Collaborative Learning: Encourage group discussions and collaborative activities to foster interaction and knowledge sharing among students. Assign group projects or case studies that require teamwork and problem-solving. 56. Demonstrations and Simulations: Use demonstrations and simulations to showcase complex concepts or processes. This can include network simulations, software simulations, or virtual environments to reinforce understanding and practical application. 57. Guest Speakers and Industry Experts: Invite guest speakers from the industry to share their experiences, insights, and the latest trends in the IT field. This can provide students with a real-world perspective and inspire them to explore various career paths. 58. Online Resources and Multimedia: Utilize online resources, interactive tutorials, and multimedia materials to supplement learning. This can include video lectures, online quizzes, virtual labs, and interactive modules. 59. Assignments and Projects: Assign individual and group projects that require students to apply their knowledge and skills to solve real-world problems or complete practical tasks. This promotes critical thinking, problem-solving, and practical application of concepts. 60. Assessments and Feedback: Conduct regular assessments, quizzes, and

	examinations to evaluate students' understanding of the course material. Provide timely and constructive feedback to help students identify areas of improvement. 61. Industry Visits and Field Trips: Organize visits to IT companies, data centers, or relevant organizations to expose students to real-world IT environments. This provides valuable industry insights and networking opportunities. 62. Online Discussion Forums and Communication Platforms: Establish online discussion forums or communication platforms where students can ask questions, share resources, and engage in discussions outside of the classroom.

Delivery Plan (Weekly Syllabus)

المنهاج الاسبوعي النظري

	Material Covered
Week 1	Introduction - Software definition - Applications and problems - Software engineering definition.
Week 2	Software Process - The conflicting and complementary goals of SWE - Water fall Mode
Week 3	Software Engineering Paradigms - Prototyping - Evolutionary development - Formal systems development
Week 4	Software Requirement - Objectives - Requirements Engineering Process - Types of Requirements - Software Requirement Specification - Software Requirement Validation
Week 5	Analysis Model - Analysis Model Types and examples

	- Formal Specifications - Formal methods - Formal Specification Languages
Week 6-7	Software Design - Design and Quality - Software Design Levels - Fundamental Design Concepts
Week 8	Functional Independence - Definition - Cohesion and its types - Coupling and its types - Effective Modular Design
Week 9-10	Design Element - Data - Architectural and procedural design - Top-Down and Bottom-Up Design - Structured Design - Transform and Transaction Mapping
Week 11	Software Testing - Definition and Objectives - Exhaustive Testing - Test case design - Software Testing Strategies
Week 12	White Box Testing - Basis path testing - Basis path method with examples - Condition testing - Data flow testing - Loop testing
Week 13	Black Box Testing - Black box testing techniques

Week 14	Software Management - Software Project definition and goal
Week 15	Project Scheduling - Resource management - Project Execution & Monitoring - Project Management Tools

Learning and Teaching Resources

مصادر التعلم والتدريس

	Text	Available in the Library?
Required Texts	The textbooks cover the fundamental concepts, principles, and practices of Software Engineering. Some popular options include " Software engineering A practitioner's approach, Third Edition, Roger S. Pressman, 2005.	**Yes**
Recommended Texts		
Websites		

MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information

معلومات المادة الدراسية

Module Title	**Routing and Switching**	**Module Delivery**

Module Aims, Learning Outcomes and Indicative Contents

أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية

Module Objectives أهداف المادة الدراسية	In this course, Networks department aims to achieve the following objectives: 1. Gain a fundamental understanding of networking concepts,

	including switches, routers, VLANs, and inter-VLAN routing. 2. Develop practical skills in configuring basic settings on switches and routers. 3. Understand and configure Spanning Tree Protocol (STP) for loop prevention in switched networks. 4. Learn about EtherChannel and link aggregation techniques for bundling multiple links. 5. Acquire knowledge of DHCP and DHCP relay agent configuration for dynamic IP address assignment. 6. Learn switch security mechanisms, including port security and VLAN security. 7. Understand the basics of OSPF (Open Shortest Path First) and configure OSPF in single and multiple areas. 8. Gain familiarity with firewalls and learn to configure basic firewall rules. 9. Learn about Access Control Lists (ACLs) and Network Address Translation (NAT) concepts and configuration. 10. Gain knowledge of WAN concepts, including different WAN technologies and protocols. 11. Review and reinforce the concepts covered throughout the course.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	Upon successful completion of this course for the Networks department, students should be able to demonstrate the following learning outcomes: 1. Understanding of Networking Fundamentals: Students will demonstrate a solid understanding of networking concepts, including switches, routers, VLANs, inter-VLAN routing, and basic network protocols. 2. Configuration and Troubleshooting Skills: Students will develop practical skills in configuring switches and routers, implementing Spanning Tree Protocol (STP), configuring EtherChannel and link aggregation, DHCP, OSPF, firewalls, ACLs, NAT, and troubleshooting network connectivity and performance issues. 3. Security Knowledge: Students will gain knowledge of switch security mechanisms, such as port security and VLAN security, as well as basic firewall configuration. 4. Proficiency in OSPF: Students will demonstrate proficiency in configuring OSPF in single and multiple areas, understanding OSPF network types, authentication, and OSPFv3. 5. Familiarity with WAN Technologies: Students will gain familiarity with different WAN technologies, such as leased lines, MPLS, VPN, and WAN protocols like PPP and HDLC. 6. Understanding of Network Services: Students will learn about important network services, including DHCP and DNS.

	7. Application of Access Control Lists (ACLs) and Network Address Translation (NAT): Students will be able to configure ACLs for traffic filtering and implement NAT for IP address translation. 8. Troubleshooting Skills: Students will develop effective troubleshooting methodologies and techniques to identify and resolve network issues. 9. Exam Readiness: Students will be prepared to take the CCNA Routing & Switching exam and demonstrate their knowledge and skills in the field of network routing and switching. Overall Competence in Networking: By the end of the course, students will have a comprehensive understanding of network routing and switching concepts, configurations, and troubleshooting, enabling them to pursue further studies or careers in network administration and engineering.
Indicative Contents المحتويات الإرشادية	The indicative contents of this course for the Networks department may include the following topics: This course covers the in-depth understanding of various networking concepts and technologies. The course begins with an introduction to networking concepts and an overview of switches and routers, followed by basic configuration concepts for switches and routers. The course then covers Virtual LAN (VLAN) concepts and how to create and configure VLANs on switches. It also covers inter-VLAN routing, VLAN trunking protocols (VTP), and VLAN connectivity. The course then goes into Spanning Tree Protocol (STP) concepts and operation, including configuring and troubleshooting STP, as well as STP enhancements (RSTP, MSTP), root bridge election, and port roles. It then covers link aggregation concepts, including configuring and testing EtherChannel, as well as Dynamic Host Configuration Protocol (DHCP) and DHCP Relay Agent configuration. The course also covers switch security mechanisms, including port security, VLAN security, and securing switch management. It discusses Open Shortest Path First (OSPF) basics and operation in a single area, OSPF areas and network types, and OSPF authentication and OSPFv3. The course then moves onto firewalls, including an introduction to firewalls and their types, as well as firewall architectures and basic configuration. It also covers Access Control Lists (ACLs) concepts and configuration, as well as Network Address Translation (NAT) concepts and configuration.

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	Learning and teaching strategies for the IT basics course for the Network department can include a combination of the following: 63. Lectures: Engage students through informative lectures that cover theoretical concepts and provide an overview of key topics. Use multimedia resources, visuals, and real-world examples to enhance understanding. 64. Hands-on Labs: Provide practical lab sessions where students can apply their knowledge and skills acquired in lectures. These labs can involve hardware assembly, software installation, network configuration, programming exercises, and troubleshooting. 65. Group Discussions and Collaborative Learning: Encourage group discussions and collaborative activities to foster interaction and knowledge sharing among students. Assign group projects or case studies that require teamwork and problem-solving. 66. Demonstrations and Simulations: Use demonstrations and simulations to showcase complex concepts or processes. This can include network simulations, software simulations, or virtual environments to reinforce understanding and practical application. 67. Guest Speakers and Industry Experts: Invite guest speakers from the industry to share their experiences, insights, and the latest trends in the IT field. This can provide students with a real-world perspective and inspire them to explore various career paths. 68. Online Resources and Multimedia: Utilize online resources, interactive tutorials, and multimedia materials to supplement learning. This can include video lectures, online quizzes, virtual labs, and interactive modules. 69. Assignments and Projects: Assign individual and group projects that require students to apply their knowledge and skills to solve real-world problems or complete practical tasks. This promotes critical thinking, problem-solving, and practical application of concepts. 70. Assessments and Feedback: Conduct regular assessments, quizzes, and examinations to evaluate students' understanding of the course material. Provide timely and constructive feedback to help students identify areas of improvement. 71. Industry Visits and Field Trips: Organize visits to IT companies, data centers, or relevant organizations to expose students to real-world IT environments. This provides valuable industry insights and networking opportunities. 72. Online Discussion Forums and Communication Platforms: Establish online discussion forums or communication platforms where students can ask questions, share resources, and engage in discussions outside of the classroom. These strategies promote active learning, practical application of knowledge, and engagement with the subject matter. They cater to different learning styles and encourage students to develop critical thinking, problem-solving, and

	communication skills necessary for success in the IT field.

Delivery Plan (Weekly Syllabus) المنهاج الاسبوعي النظري	
	Material Covered
Week 1	• Switch Basic Configuration
Week 2	• Router Basic Configuration
Week 3-4	• VLAN Configuration
Week 5	• Connecting VLANs
Week 6	• Spanning Tree Protocol (STP) Part 1
Week 7	• DHCP and DHCP Relay Agent
Week 8	• DHCP IPv6
Week 9	• Mid Term
Week 10	• Switch Security 1
Week11	• Switch Security 2
Week 12	• Single Area OSPF
Week 13	• OSPF Part 2
Week 14	• WAN Concepts
Week 15	• Review
Week 16	• Preparatory week before the final Exam

Delivery Plan (Weekly Lab. Syllabus) المنهاج الاسبوعي للمختبر	
	Material Covered
Week 1-2	4. Lab: Configuring basic settings on a switch and routers, such as hostname, passwords, IP addresses, and interface descriptions. Verify connectivity

	between devices. ○
Week 3-4	12. Creating and configuring VLANs on switches, configuring VLAN trunking, implementing VLAN tagging. ○
Week 5-6	○ Lab: Configuring STP, troubleshooting common STP issues.
Week 7-8	○ DHCP and DHCP Relay Agent
Week 9	○ Mid Term
Week 10-11	○ Switch Security
Week 12	○ Single Area OSPF
Week 13	○ OSPF Part 2
Week 14	○ ACL and NAT
Week 15	○ Review

Learning and Teaching Resources مصادر التعلم والتدريس		
	Text	**Available in the Library?**
Required Texts	TCP/IP PROTOCOL SUITE	
Recommended Texts		
Websites		

MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information معلومات المادة الدراسية		
Module Title	**Visual Programming**	**Module Delivery**

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Objectives أهداف المادة الدراسية	The Visual programming course for the Networks department aims to achieve the following objectives: 1. Understanding Visual Programming Concepts: Gain a solid understanding of the fundamental concepts and principles of visual programming, including visual representations, event-driven programming, and graphical user interfaces (GUIs). 2. Proficiency in Visual Programming Tools: Develop proficiency in using popular visual programming tools and environments such as Scratch, Blockly, or visual programming languages like Visual Basic, Python with Tkinter, or App Inventor. 3. GUI Design and Development: Learn to design and develop user-friendly graphical user interfaces (GUIs) using visual programming tools, including layout design, component placement, and interactive elements. 4. Event-Driven Programming: Understand the concepts of event-driven programming and learn how to create event handlers and respond to user input and system events in visual programming environments. 5. Algorithmic Thinking and Problem Solving: Enhance algorithmic thinking and problem-solving skills by developing logical and computational thinking through visual programming challenges and projects. 6. Integration of Multimedia and Sensors: Explore the integration of multimedia elements such as images, audio, and video, as well as sensor inputs like motion, sound, and touch, into visual programming projects. 7. Collaboration and Teamwork: Foster collaboration and teamwork skills through group projects that involve designing, developing, and presenting visual programming applications. 8. Debugging and Troubleshooting: Develop the ability to identify and resolve errors and bugs in visual programming code through effective debugging and troubleshooting techniques. 9. Creativity and Innovation: Encourage creativity and innovation by allowing students to explore and create interactive and visually appealing applications using visual programming tools. 10. Ethical and Responsible Use of Visual Programming: Promote ethical and responsible use of visual programming by emphasizing issues such as privacy, security, intellectual property, and social impact. 11. Project Management and Documentation: Gain experience in project management by planning, organizing, and documenting visual programming projects, including requirements gathering, design, implementation, testing, and documentation. 12. Continuous Learning and Adaptation: Develop a mindset of continuous learning and adaptation to keep up with emerging trends and advancements in visual programming and related technologies.
Module Learning Outcomes	Upon successful completion of the Visual programming course for the Networks department, students should be able to demonstrate the following

مخرجات التعلم للمادة الدراسية	learning outcomes: 3. Proficiency in Visual Programming Tools: Students should be able to effectively use visual programming tools and environments to create functional and visually appealing applications with user-friendly interfaces. 4. GUI Design and Development: Students should be capable of designing and developing graphical user interfaces (GUIs) using visual programming techniques, including layout design, component placement, and interactive features. 5. Event-Driven Programming: Students should understand the concepts of event-driven programming and be able to create event handlers and respond to user input and system events in visual programming environments. 6. Algorithmic Thinking and Problem Solving: Students should demonstrate the ability to apply algorithmic thinking and problem-solving skills to develop logical and computational solutions to programming challenges within the visual programming paradigm. 7. Integration of Multimedia and Sensors: Students should be able to integrate multimedia elements, such as images, audio, and video, as well as sensor inputs like motion, sound, and touch, into their visual programming projects. 8. Collaboration and Teamwork: Students should have experience working collaboratively in teams to design, develop, and present visual programming applications, demonstrating effective communication, cooperation, and shared responsibility. 9. Debugging and Troubleshooting: Students should possess the skills to identify and resolve errors and bugs in visual programming code through effective debugging and troubleshooting techniques. 10. Creativity and Innovation: Students should showcase creativity and innovation by creating unique and interactive applications that go beyond basic requirements, incorporating novel ideas, design elements, or features. 11. Ethical and Responsible Use of Visual Programming: Students should exhibit an understanding of ethical considerations related to visual programming, including privacy, security, intellectual property, and the social impact of their applications. 12. Project Management and Documentation: Students should demonstrate the ability to plan, organize, and document visual programming projects, including requirements gathering, design documentation, code commenting, and user instructions. 13. Continuous Learning and Adaptation: Students should exhibit a mindset of continuous learning and adaptation, being aware of emerging trends and advancements in visual programming and related technologies and being capable of independently learning and exploring new tools and concepts.
Indicative Contents المحتويات الإرشادية	The indicative contents of the Visual programming course for the computer department may include the following topics: 55. Introduction to Visual Programming

- Overview of visual programming concepts and benefits
- Introduction to visual programming tools and environments
- Basic elements and features of visual programming interfaces

56. GUI Design and Layout
    - Principles of graphical user interface (GUI) design
    - Layout managers and component placement
    - Styling and customization of GUI elements
57. Event-Driven Programming
    - Introduction to event-driven programming paradigm
    - Handling user input events (e.g., button clicks, mouse movements)
    - Responding to system events (e.g., window events, timer events)
58. Data Manipulation and Visualization
    - Working with data structures and variables in visual programming
    - Displaying data using charts, graphs, and other visualization techniques
    - Interacting with data through input forms and user controls
59. Multimedia Integration
    - Incorporating images, audio, video, and animations into visual programming projects
    - Manipulating multimedia elements using visual programming tools
    - Creating interactive multimedia applications
60. Animation and Game Development
    - Introduction to animation concepts in visual programming
    - Creating animations and transitions
    - Developing simple games using visual programming techniques
61. Database Integration
    - Connecting visual programming applications to databases
    - Retrieving and manipulating data from a database using visual programming tools
    - Creating interactive forms for data entry and retrieval
62. Web and Mobile Application Development
    - Introduction to web and mobile application development using visual programming
    - Creating web pages or mobile app interfaces with visual programming tools
    - Integrating web services or mobile device features into visual programming applications
63. Advanced Visual Programming Concepts
    - Advanced GUI design techniques (e.g., drag-and-drop, custom controls)
    - Multi-threading and concurrency in visual programming
    - Extending visual programming functionality with plugins or extensions
64. Project Development
    - Planning, designing, and implementing a visual programming project
    - Applying learned concepts and techniques to develop a substantial application

	○ Iterative development, testing, and debugging of the project 65. User Experience (UX) Design and Usability ○ Introduction to UX design principles and usability considerations ○ Conducting user testing and incorporating user feedback into visual programming projects ○ Enhancing the user experience through effective design choices 66. Deployment and Distribution ○ Packaging and distributing visual programming applications for different platforms ○ Considerations for deployment on web, desktop, or mobile platforms ○ App store guidelines and submission processes

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	Learning and teaching strategies for the Visual programming course for the 1. Hands-on Projects: Encourage students to actively engage in hands-on projects throughout the course. Assign programming tasks and projects that allow students to apply the visual programming concepts they have learned. Provide them with real-world scenarios and challenges to solve using visual programming tools. 2. Interactive Demos and Examples: Use interactive demos and examples to demonstrate the application of visual programming concepts. Show step-by-step development of applications, highlighting key features and techniques. Encourage students to experiment with the provided examples and modify them to gain a deeper understanding. 3. Collaborative Learning: Promote collaborative learning by assigning group projects or pairing students for programming tasks. Encourage students to discuss and share ideas, troubleshoot issues together, and provide feedback to their peers. This fosters teamwork and enhances problem-solving skills through collective effort. 4. Practice and Experimentation: Assign regular coding exercises and practice sessions to reinforce learning. Provide a variety of problems and challenges to tackle, allowing students to practice different aspects of visual programming. Encourage experimentation and creativity by giving them the freedom to explore different approaches and solutions. 5. Code Review and Feedback: Incorporate code review sessions where students can present their projects and receive feedback from the instructor and their peers. Provide constructive criticism to help students improve their coding style, design choices, and adherence to best practices in visual programming. 6. Online Resources and Tutorials: Share supplementary online resources, tutorials, and documentation related to visual programming tools and concepts. Point students to helpful websites, forums, and video tutorials where they can find additional learning materials and examples to

	deepen their understanding. 7. Mini-Projects and Challenges: Introduce mini-projects and coding challenges that require students to think creatively and solve specific problems using visual programming. These smaller-scale projects allow students to focus on specific skills or concepts and provide opportunities for self-assessment and self-improvement. 8. Reflective Learning: Encourage students to reflect on their learning progress and experiences. Assign periodic reflection papers or journal entries where they can express their thoughts, challenges faced, and lessons learned while working on visual programming projects. This promotes metacognition and helps students identify areas of improvement. 9. Guest Speakers and Industry Insights: Invite guest speakers from the industry who have expertise in visual programming or have utilized visual programming tools in their work. They can share real-world applications and insights, providing students with a broader perspective on the relevance and practical applications of visual programming. 10. Documentation and Documentation Review: Emphasize the importance of documenting code and maintaining clear, well-structured project documentation. Teach students how to write effective comments, documentation, and README files. Conduct documentation review sessions to assess their ability to communicate their code and project effectively.

Delivery Plan (Weekly Syllabus) المنهاج الاسبوعي النظري	
	Material Covered
Week 1	Week 1: Introduction to Visual Programming • Overview of visual programming concepts and tools • Getting familiar with the visual programming environment
Week 2	Week 2: GUI Design and Layout • Exploring layout managers and component placement • Creating simple user interfaces
Week 3	Week 3: Event-Driven Programming • Understanding event-driven programming paradigm • Handling user input events and system events
Week 4	Week 4: Data Manipulation and Visualization • Working with data structures and variables in visual programming

	• Displaying data using charts, graphs, or other visualization techniques
Week 5	Week 5: Multimedia Integration • Incorporating images, audio, video, and animations into projects • Manipulating multimedia elements using visual programming tools
Week 6	Week 6: Animation and Game Development • Creating animations and transitions • Developing simple games using visual programming techniques
Week 7	Week 7: Database Integration • Connecting visual programming applications to databases • Retrieving and manipulating data using visual programming tools
Week 8	Week 8: Web and Mobile Application Development • Introduction to web and mobile app development with visual programming • Creating web pages or mobile app interfaces
Week 9	Week 9: Advanced Visual Programming Concepts • Exploring advanced GUI design techniques • Multithreading and concurrency in visual programming
Week 10	Week 10: Project Development (Part 1) • Planning and designing a visual programming project • Implementing core features of the project
Week 11	Week 11: Project Development (Part 2) • Continuing project implementation • Testing and debugging the project
Week 12	Week 12: User Experience (UX) Design and Usability • Introduction to UX design principles and usability considerations • Enhancing the user experience of visual programming projects
Week 13	Week 13: Deployment and Distribution • Packaging and distributing visual programming applications • Considerations for deploying on different platforms
Week 14	Week 14: Project Refinement and Review • Finalizing project implementation

	• Conducting code review and project review sessions
Week 15	Week 15: Final Project Presentation and Evaluation • Presenting visual programming projects to the class • Evaluation and feedback on the projects
Week 16	**Preparatory week before the final Exam**

Delivery Plan (Weekly Lab. Syllabus) المنهاج الاسبوعي للمختبر	
	Material Covered
Week 1	Lab 1: Introduction to Visual Programming Tools ○ Familiarization with the visual programming environment ○ Exploring the basic features and functionalities of the chosen visual programming tool ○ Creating a simple "Hello World" application
Week 2	Lab 2: GUI Design and Layout ○ Applying layout managers to create a responsive user interface ○ Adding components and arranging them within the interface ○ Customizing the appearance and styling of GUI elements
Week 3	Lab 3: Event-Driven Programming ○ Implementing event handlers for user interactions (e.g., button clicks, mouse events) ○ Responding to system events (e.g., window events, timer events) ○ Creating interactive applications with event-driven programming
Week 4	Lab 4: Data Manipulation and Visualization ○ Working with data structures (e.g., arrays, lists) in visual programming ○ Displaying data using charts, graphs, or other visualization techniques ○ Creating interactive data-driven applications
Week 5	Lab 5: Multimedia Integration ○ Incorporating multimedia elements (e.g., images, audio, video) into projects ○ Manipulating multimedia assets using visual programming tools

	○ Developing interactive multimedia applications
Week 6	Lab 6: Animation and Game Development ○ Creating animations and transitions using visual programming tools ○ Developing a simple game using visual programming techniques ○ Implementing game mechanics and user interactions
Week 7	Lab 7: Project Development and Review ○ Applying learned concepts to develop a small-scale visual programming project ○ Testing and debugging the project ○ Presenting the project and receiving feedback from peers and the instructor

Learning and Teaching Resources مصادر التعلم والتدريس		
	Text	**Available in the Library?**
Required Texts	Microsoft Visual C# Step by Step, 10th Edition, By John Sharp Microsoft Press, 2022	
Recommended Texts		
Websites		

MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information معلومات المادة الدراسية		
Module Title	Website design and programming 2	**Module Delivery**

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية

Module Objectives أهداف المادة الدراسية	The course objectives for Web Development 2 may include: 1. Advanced Web Technologies: Introduce students to advanced web technologies and frameworks, such as server-side scripting languages (e.g., PHP, Python), content management systems (e.g., WordPress), and client-side frameworks (e.g., React, Angular). 2. Database Integration: Teach students how to integrate databases into web applications, including database design, querying, and data manipulation using SQL. Focus on concepts such as data modeling, normalization, and database connectivity. 3. Dynamic Web Development: Enable students to build dynamic web applications by integrating server-side scripting languages with client-side technologies. Cover topics like session management, form handling, user authentication, and data validation. 4. Web Security: Raise awareness about common web security vulnerabilities and techniques to secure web applications. Teach students about secure coding practices, input validation, authentication mechanisms, and protection against common attacks like cross-site scripting (XSS) and SQL injection. 5. Web Performance Optimization: Explore techniques to improve the performance and efficiency of web applications. Cover topics like caching, code minification, image optimization, and front-end optimization techniques to enhance the user experience. 6. Responsive Web Design: Introduce students to the principles of responsive web design and teach them how to create websites that adapt and display well on different devices and screen sizes. Cover CSS frameworks, media queries, and techniques for creating responsive layouts. 7. Web Accessibility: Emphasize the importance of creating web applications that are accessible to users with disabilities. Teach students about accessibility standards, techniques for implementing accessible features, and the use of assistive technologies. 8. Project Development: Provide opportunities for students to work on larger-scale web development projects. Encourage collaborative project work, where students can apply their knowledge and skills to build real-world web applications. 9. Industry Practices and Emerging Trends: Keep students updated with current industry practices and emerging trends in web development. Introduce them to topics like progressive web apps, single-page applications, API integrations, and the use of modern tools and frameworks. 10. Professional Development: Foster professional skills by promoting effective communication, teamwork, project management, and problem-solving abilities within the context of web development projects. These objectives aim to equip students with the necessary knowledge, skills, and techniques to become proficient web developers capable of building dynamic and secure web applications using the latest technologies and industry best practices.
Module Learning Outcomes	The learning outcomes for Web Development 2 may include:

مخرجات التعلم للمادة الدراسية	1. Advanced Web Technologies: Demonstrate proficiency in using advanced web technologies and frameworks, such as server-side scripting languages, content management systems, and client-side frameworks, to develop robust and scalable web applications. 2. Database Integration: Apply database integration techniques to create dynamic web applications, including database design, querying, and data manipulation. Develop skills in working with databases and understanding the importance of efficient data management. 3. Dynamic Web Development: Build dynamic web applications by integrating server-side scripting languages with client-side technologies. Implement features like session management, form handling, user authentication, and data validation to create interactive and responsive web experiences. 4. Web Security: Identify and mitigate common web security vulnerabilities. Implement secure coding practices, employ authentication mechanisms, and protect against common attacks like cross-site scripting (XSS) and SQL injection to ensure the security of web applications. 5. Web Performance Optimization: Optimize the performance of web applications by implementing techniques such as caching, code minification, and front-end optimization. Improve website loading speed and user experience through efficient resource management. 6. Responsive Web Design: Create responsive web designs that adapt to different devices and screen sizes. Develop skills in using CSS frameworks, media queries, and responsive layout techniques to ensure consistent and visually appealing experiences across multiple platforms. 7. Web Accessibility: Design and develop web applications that are accessible to users with disabilities. Apply accessibility standards and techniques to ensure equal access and usability for all users, considering factors such as screen readers, keyboard navigation, and alternative text for images. 8. Project Development: Collaborate with a team to plan, design, and implement larger-scale web development projects. Apply project management principles, communicate effectively, and work collaboratively to deliver high-quality web applications. 9. Industry Practices and Emerging Trends: Stay informed about current industry practices and emerging trends in web development. Demonstrate awareness of technologies, tools, and frameworks used in the industry, and adapt to changing demands and advancements in the field. 10. Professional Growth: Demonstrate professionalism in web development by effectively communicating ideas, solving problems, and adapting to new challenges. Continuously improve skills and stay updated with industry advancements through self-directed learning and professional development opportunities. These learning outcomes aim to equip students with the knowledge, skills, and abilities to become proficient web developers who can create dynamic, secure, and user-friendly web applications using advanced technologies and industry best practices.

Indicative Contents المحتويات الإرشادية	The indicative contents for Web Development 2 may include: 1. Introduction to Server-Side Scripting: Overview of server-side scripting languages such as PHP, Python, or Node.js. Understanding the server-side architecture and the role of server-side languages in web development. 2. Database Integration: Exploring advanced database integration techniques using SQL or NoSQL databases. Topics may include database design, advanced querying, data manipulation, and database security. 3. Content Management Systems (CMS): Introduction to popular CMS platforms like WordPress, Drupal, or Joomla. Understanding the architecture, theme development, plugin customization, and content management using CMS. 4. Web Frameworks: Introduction to popular web frameworks such as Ruby on Rails, Django, or Laravel. Exploring the features, MVC architecture, routing, database integration, and rapid development using web frameworks. 5. RESTful API Development: Understanding the concepts of RESTful APIs and their role in web development. Building and consuming RESTful APIs using popular frameworks or libraries like Express.js or Flask. 6. Authentication and Authorization: Implementing user authentication and authorization mechanisms in web applications. Topics may include user registration, login/logout functionality, password hashing, and role-based access control. 7. Web Security Best Practices: Exploring advanced web security concepts and best practices. Topics may include handling user input securely, preventing common vulnerabilities like CSRF and XSS attacks, and implementing secure coding practices. 8. Web Performance Optimization: Techniques for optimizing the performance of web applications. Topics may include caching, asynchronous loading, minification, image optimization, and front-end performance best practices. 9. Responsive Web Design: Advanced concepts in responsive web design. Exploring responsive frameworks, media queries, responsive images, and techniques for creating mobile-friendly and adaptive web layouts. 10. Testing and Debugging: Strategies for testing and debugging web applications. Topics may include unit testing, integration testing, browser debugging tools, and error handling techniques. 11. Version Control and Collaboration: Introduction to version control systems like Git and their role in collaborative web development. Understanding branching, merging, resolving conflicts, and collaborative development workflows. 12. Project Development: Working on a larger-scale web development project in a team environment. Applying project management principles, agile development methodologies, and effective communication and collaboration skills.

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	The learning and teaching strategies for Web Development 2 can include: 1. Lectures: The instructor delivers lectures to introduce new concepts, explain theoretical aspects, and provide examples and case studies related to web development. 2. Hands-on Coding: Students engage in hands-on coding exercises and projects to apply their knowledge and skills in building web applications. They can work individually or in groups to develop real-world projects, implementing the concepts learned during the course. 3. Code Review and Feedback: Students participate in code reviews where they share their code with peers and receive feedback. This promotes collaboration, peer learning, and improvement of coding practices. 4. Practical Examples and Demonstrations: The instructor demonstrates practical examples and showcases real-world applications to illustrate the concepts and techniques in web development. This helps students visualize the application of the learned concepts. 5. Discussion and Debate: Students engage in discussions and debates on topics related to web development, such as emerging trends, best practices, and ethical considerations. This encourages critical thinking, problem-solving, and the exploration of different perspectives. 6. Guest Speakers and Industry Experts: Inviting guest speakers and industry experts to share their experiences and insights in web development can provide students with valuable industry perspectives and practical knowledge. 7. Workshops and Tutorials: Conducting workshops and tutorials where students can work on specific web development tasks, solve problems, and learn new tools and technologies. These sessions can be interactive and allow students to receive guidance and support from the instructor. 8. Online Resources and Self-Study: Providing access to online resources, tutorials, documentation, and coding exercises to encourage self-study and exploration. Students can deepen their understanding of web development concepts and technologies at their own pace. 9. Project-based Learning: Assigning individual or group projects that require students to design, develop, and deploy web applications. This allows them to apply their knowledge, practice problem-solving skills, and gain hands-on experience in real-world scenarios. 10. Assessments and Feedback: Conducting regular assessments, quizzes, and assignments to evaluate students' understanding of the concepts and their ability to apply them. Providing timely feedback helps students identify areas of improvement and reinforce their learning. 11. Collaborative Learning: Encouraging collaboration among students through group work, pair programming, or collaborative coding sessions. This fosters teamwork, communication skills, and the sharing of knowledge and expertise. 12. Industry Case Studies: Presenting case studies of successful web development projects and applications in various industries. This helps students understand the practical application of web development skills

	and the challenges faced in real-world scenarios. These strategies aim to create an engaging and interactive learning environment that promotes active participation, practical application of knowledge, and continuous learning in the field of web development.

Delivery Plan (Weekly Syllabus) المنهاج الاسبوعي النظري	
	Material Covered
Week 1	: Introduction to Server-Side Programming • Overview of server-side programming languages (e.g., PHP, Python, Node.js) • Setting up a development environment with a server-side language and a web server
Week 2-3	Database Integration • Introduction to database systems (e.g., MySQL, PostgreSQL) • Connecting a web application to a database • Querying and manipulating data using SQL
Week 4-5	Advanced JavaScript and DOM Manipulation • JavaScript libraries and frameworks for front-end development (e.g., React, Angular) • Advanced DOM manipulation techniques • Handling asynchronous operations using AJAX and promises
Week 6-7	Web Application Security • Common web vulnerabilities (e.g., Cross-Site Scripting, SQL injection) • Techniques for securing web applications • Implementing user authentication and authorization
Week 8-9	Server-Side Frameworks • Introduction to popular server-side frameworks (e.g., Laravel, Django, Express.js) • Building dynamic web applications using a framework • Implementing RESTful APIs
Week 10-11	Version Control and Collaboration • Introduction to version control systems (e.g., Git) • Collaborative web development using Git and GitHub • Deployment strategies for web applications

Week 12-13	Testing and Debugging • Writing unit tests for web applications • Debugging techniques for identifying and fixing issues • Performance optimization and code profiling
Week 14-15	Project Development and Presentation • Applying the learned concepts to develop a complete web application • Project planning, development, and documentation • Presenting and demonstrating the web application

Delivery Plan (Weekly Lab. Syllabus) المنهاج الاسبوعي للمختبر	
	Material Covered
Week 1	Review of Web Development Basics • Refreshing HTML, CSS, and JavaScript concepts • Recap of responsive web design principles
Week 2	Advanced CSS Techniques • Implementing CSS animations and transitions • Using CSS preprocessors (e.g., Sass, Less) • Creating CSS frameworks and libraries
Week 3	Advanced JavaScript Concepts • Exploring advanced JavaScript topics (e.g., closures, prototypes) • Working with JavaScript libraries (e.g., jQuery, Lodash) • Building modular JavaScript code using modules and namespaces
Week 4	Single-Page Applications (SPA) • Introduction to SPA architecture and frameworks (e.g., Angular, React, Vue) • Building a simple SPA using a chosen framework • Routing and navigation in SPAs
Week 5	Server-Side Scripting with Node.js 15. Setting up a Node.js development environment 16. Writing server-side JavaScript code using Node.js 17. Implementing server-side functionality and APIs

Week 6	Database Integration • Introduction to database management systems (e.g., MySQL, MongoDB) • Interacting with databases using server-side scripting (e.g., CRUD operations) • Implementing data persistence in web applications
Week 7	Authentication and Authorization • Implementing user registration and login functionality • Exploring authentication and authorization techniques (e.g., sessions, tokens) • Securing web applications against common vulnerabilities (e.g., cross-site scripting, SQL injection)
Week 8	API Development • Designing and implementing RESTful APIs • Handling API requests and responses • Documentation and testing of APIs
Week 9	Web Performance Optimization • Techniques for optimizing web page load times • Implementing caching strategies • Analyzing and improving website performance using tools (e.g., PageSpeed Insights, Lighthouse)
Week 10	Advanced Front-End Frameworks • Exploring advanced features and components of front-end frameworks (e.g., Angular, React) • Building complex web applications with front-end frameworks •
Week 11-14	Web Project Development • Applying the concepts learned to develop a complex web project • Planning, designing, and implementing a dynamic web application • Testing, debugging, and optimizing the web project
Week 15	Project Presentation and Evaluation

Learning and Teaching Resources

مصادر التعلم والتدريس

	Text	Available in the Library?
Required Texts		No

Recommended Texts		
Websites		