# Mosul University

# جامعة الموصل

*Bachelor's degree (B.Sc.) – Software*

بكالوريوس علوم  - البرمجيات

# Table of Contents

# 1. Overview

This catalogue is about the courses (modules) given by the program of software department for the services given to university and society in the field of software analysis and design with implementation for large and medium software's, to gain the Bachelor of software Science degree. The program delivers 50) Modules with (5950) total student workload hours and 238 total ECTS. The module delivery is based on the Bologna Process.

نظره عامه

السعي لان يكون قسم هندسة البرمجيات منبرا علميا من خلال الخدمات التي يقدمها للجامعة والمجتمع في مجال تحليل وتصميم وتنفيذ البرمجيات المتوسطة والكبيرة   للحصول على درجة بكالوريوس العلوم  في البرمجيات. يقدم البرنامج (50) مادة دراسية، على سبيل المثال، مع (5950) إجمالي ساعات حمل الطالب و  240 إجمالي وحدات أوروبية. يعتمد تقديم المواد الدراسية على عملية بولونيا.

# 2. Undergraduate Courses 2023-2024

**Module 1**

| Code | Course/Module Title | ECTS | Semester |
|---|---|---|---|
| UOMCSSW1101 | **Democracy and Human Rights** | 2 | 1 |
| **Class (hr/w)** | **Lect/Lab./Prac./Tutor** | **SSWL (hr/sem)** | **USWL (hr/w)** |
| 2 | | 33 | 17 |
| **Description** | | | |
| This section includes a description of the module, 100-150 words | | | |

**Module 2**

| Code | Course/Module Title | ECTS | Semester |
|---|---|---|---|
| UOMCSSW1102 | **Logic design** | 8 | 1 |
| **Class (hr/w)** | **Lect/Lab./Prac./Tutor** | **SSWL (hr/sem)** | **USWL (hr/w)** |
| 2 | 2+1 | 78 | 122 |

| Description |
|---|

This module introduces the concepts of the design and implementation of digital circuits. Laboratory experiments will be used to reinforce the theoretical concepts discussed in lectures. The lab experiments will involve the design and implementation of digital circuits. Emphasis is on the use computer aided tools in the design, simulation, and testing of digital circuits.

Course Objectives: The aim of the module is to introduce to the students the topics that include combinational and sequential circuit analysis and design, digital circuit design optimization methods using random logic gates, multiplexers, decoders, registers, counters and programmable logic arrays.

**Module 3**

| Code | Course/Module Title | ECTS | Semester |
|---|---|---|---|
| UOMCSSW1103 | **Computer** | 3 | 1 |
| **Class (hr/w)** | **Lect/Lab./Prac./Tutor** | **SSWL (hr/sem)** | **USWL (hr/w)** |
| 1 | 2 | 48 | 27 |
| **Description** | | | |

Basic computer skills refer to your ability to perform fundamental tasks on a computer. This can include understanding certain software, applications, programs and tools. Computer skills fall into two categories: software and hardware. Hardware refers to the computer's operation, whereas software involves the use of various programs and applications. Software skills help you to efficiently use computer programs and applications. There are some software skills that employers may consider as prerequisites to employment.

Introduction to computer system, Computer Generations Parts of Computer System (Hardware, Software, User, Data), Types of Computers, Computer's Performance Factors Computer Hardware, The Main Components of Computers (CPU, Memory, I/O) Computer memory, Representing Data in the memory, Memory Capacity Measurements, Memory Process, haracteristic of the Memory,Types of memory Computer Software Software Types Application Software System Software Programming Languages, Types of Programming Languages, Compilers and Interpreters. Operating Systems, Operating System Objectives, Functions of Operating Systems, Types of Operating Systems Software Interface Types of Software Interface Accessibility Options Computer Network Types of Networks Network Topology Network Hardware Data Communication, Characteristics of Data Communication, Components of Data Communication, Data Representation, Communication Protocols, Data Transmission Modes Digital and Analog Transmission Asynchronous and Synchronous Transmission Viruses Types of Viruses Protection Against Viruses Software Copyright, Licensing, Type of License Agreements Software Piracy Types of Software Piracy Plagiarism Site License

**Module 4**

| Code | Course/Module Title | ECTS | Semester |
|---|---|---|---|
| UOMCSSW1104 | **Discrete Structures** | 4 | 1 |
| **Class (hr/w)** | **Lect/Lab./Prac./Tutor** | **SSWL (hr/sem)** | **USWL (hr/w)** |
| 2 | | 33 | 67 |
| **Description** | | | |

Definition to discrete structure Biconditional statement, Type of statements Algebra of propositions Laws of Algebra  propositions Define of Predicates logic Define of quantifiers and examples Sets theory Type of set operation Venn diagram in details Laws of set operation and type Define of Cartesian product types of relations Graphs of relation, construct the relation How to solve laws of relations Other approaches to computability.

The course aims to study discontinuous structures in terms of the use of algebraic laws, diagrams and shapes to reach an easy and clear way for students to solve all issues related to the subject of discontinuous structures. In addition to studying quantifiers and predicates logic and studying different groups, theories and schemes so that the student can solve the duties required of him and the exercises with simplicity, ease and clarity.

**Module 5**

| Code | Course/Module Title | ECTS | Semester |
|---|---|---|---|
| UOMCSSW1105 | **Algorithms and Structured Programming (1)** | 8 | 1 |
| **Class (hr/w)** | **Lect/Lab./Prac./Tutor** | **SSWL (hr/sem)** | **USWL (hr/w)** |
| 2 | 2+1 | 78 | 122 |
| **Description** | | | |

This course provides an introduction to mathematical modeling of computational problems. It covers the common algorithms, algorithmic paradigms, and programming structures used to solve these problems. The course emphasizes the relationship between algorithms and programming, and introduces basic performance measures and analysis techniques for these problems. Converting algorithm into c++ programming language.

Teaching the student how to plan to write a program to solve a mathematical dilemma and identify the steps of the basic program from entering, processing and printing the results. The answer. The processing consists, according to the question, of making a decision i(f) or iterating (for).

**Module 6**

| Code | Course/Module Title | ECTS | Semester |
|---|---|---|---|
| UOMCSSW1202 | **Mathematics** | 5 | 1 |
| **Class (hr/w)** | **Lect/Lab./Prac./Tutor** | **SSWL (hr/sem)** | **USWL (hr/w)** |
| 2 | 1 | 48 | 77 |
| **Description** | | | |

This section includes a description of the module, 100-150 words

**Module 6**

| Code | Course/Module Title | ECTS | Semester |
|---|---|---|---|
| UOMCSSW1106 | **Arabic language** | 1 | 1 |
| **Class (hr/w)** | **Lect/Lab./Prac./Tutor** | **SSWL (hr/sem)** | **USWL (hr/w)** |
| 1 | 1 | 18 | 7 |
| **Description** | | | |
| This section includes a description of the module, 100-150 words | | | |

**Module 7**

| Code | Course/Module Title | ECTS | Semester |
|---|---|---|---|
| UOMCSSW1201 | **Computer Skills (2)** | 8 | 2 |
| **Class (hr/w)** | **Lect/Lab./Prac./Tutor** | **SSWL (hr/sem)** | **USWL (hr/w)** |
| 1 | 3+2 | 78 | 122 |
| **Description** | | | |

Basic computer skills refer to your ability to perform fundamental tasks on a computer. This can include understanding certain software, applications, programs and tools. Computer skills fall into two categories: software and hardware. Hardware refers to the computer's operation, whereas software involves the use of various programs and applications. Software skills help you to efficiently use computer programs and applications. There are some software skills that employers may consider as prerequisites to employment

Introduction to computer system, Computer Generations Parts of Computer System (Hardware, Software, User, Data), Types of Computers, Computer's Performance Factors Computer Hardware, The Main Components of Computers (CPU, Memory, I/O) Computer memory, Representing Data in the memory, Memory Capacity Measurements, Memory Process, haracteristic of the Memory,Types of memory Computer Software Software Types Application Software System Software Programming Languages, Types of Programming Languages, Compilers and Interpreters. Operating Systems, Operating System Objectives, Functions of Operating Systems, Types of Operating Systems Software Interface Types of Software Interface Accessibility Options Computer Network Types of Networks Network Topology Network Hardware Data Communication, Characteristics of Data Communication, Components of Data Communication, Data Representation, Communication Protocols, Data Transmission Modes Digital and Analog Transmission Asynchronous and Synchronous Transmission Viruses Types of Viruses Protection Against Viruses Software Copyright, Licensing, Type of License Agreements Software Piracy Types of Software Piracy Plagiarism Site License.

**Module 8**

| Code | Course/Module Title | ECTS | Semester |
|---|---|---|---|
| UOMCSSW1202 | **Mathematics** | 4 | 2 |
| **Class (hr/w)** | **Lect/Lab./Prac./Tutor** | **SSWL (hr/sem)** | **USWL (hr/w)** |
| 1 | 2 | 33 | 67 |
| **Description** | | | |
| This section includes a description of the module, 100-150 words | | | |

**Module 9**

| Code | Course/Module Title | ECTS | Semester |
|---|---|---|---|
| UOMCSSW1203 | **Algorithms and Structured Programming (2)** | 8 | 2 |
| **Class (hr/w)** | **Lect/Lab./Prac./Tutor** | **SSWL (hr/sem)** | **USWL (hr/w)** |
| 1 | 3+2 | 78 | 122 |
| **Description** | | | |

The course introduces the students to the basic concepts and techniques of developing programs for problem solving. C++ programming methodology is used throughout the course to teach the fundamentals of programming. In this course, students learn how to apply an integrated program development tool to design, implement, test, debug, and document programs. It establishes the foundation on which students are able to develop application programs in different high-level programming languages such as Java and C++. Aims This course is aimed at introducing students to the subject of computer programming and enabling them to develop computer programs for problem solving at a basic level. and using numeric and literal arrays (1D ,2D array and string ) or structure and files.

**Module 10**

| Code | Course/Module Title | ECTS | Semester |
|---|---|---|---|
| UOMCSSW1204 | **English Language (1)** | 2 | 2 |
| **Class (hr/w)** | **Lect/Lab./Prac./Tutor** | **SSWL (hr/sem)** | **USWL (hr/w)** |
| 1 | 1 | 33 | 17 |
| **Description** | | | |

The course consists of conversations in English in the field of computers (information technology), where the student receives all the information about the computer and at the same time learns and trains the correct pronunciation in this language, as there are conversations between people about everything related to information technology. Contents: English for IT part1 Working in the IT / IT jobs and duties + listen, Working in the IT / IT origination + listen,Working in the IT / IT work rules + listen,Working in the IT / meeting + listening Working in the IT / agenda + listen,Computer systems / computer hardware + listen, Computer systems / computer Software + listen,Computer systems / work with computers + listen, IT Systems / system specifications + listen, IT Systems / GUI operations + listen, Data communications/ Email + listen.

**Module 11**

| Code | Course/Module Title | ECTS | Semester |
|---|---|---|---|
| UOMCSSW1205 | **Computer Organization** | 6 | 2 |
| **Class (hr/w)** | **Lect/Lab./Prac./Tutor** | **SSWL (hr/sem)** | **USWL (hr/w)** |
| 1 | 2+2 | 63 | 87 |
| **Description** | | | |

Computer organization is the knowing, What the functional components of a computer are, how they work and how their performance is measured and optimized. Computer Organization refers to the level of abstraction above

the digital logic level, but below the operating system level

**Module 12**

| Code | Course/Module Title | ECTS | Semester |
|---|---|---|---|
| UOMCSSW1206 | **Introduction of Statistics and Probability** | 2 | 2 |
| **Class (hr/w)** | **Lect/Lab./Prac./Tutor** | **SSWL (hr/sem)** | **USWL (hr/w)** |
| 1 | 1 | 33 | 17 |
| **Description** | | | |

This course provides an elementary introduction to probability and statistics with applications. Topics include: basic probability models; combinatorics; random variables; discrete and continuous probability distributions; statistical estimation and testing; confidence intervals; and an introduction to linear regression.

**Module 13**

| Code | Course/Module Title | ECTS | Semester |
|---|---|---|---|
| UOMCSSW2101 | **Object Oriented Programming (1)** | 8 | 3 |
| **Class (hr/w)** | **Lect/Lab./Prac./Tutor** | **SSWL (hr/sem)** | **USWL (hr/w)** |
| 2 | 3+2 | 78 | 122 |
| **Description** | | | |

Teaching students:
- How to write readable, reusable, and modular code.
- Fundamental object-oriented programming concept.
- Apply OOP concepts to your Python code.

Mastering Python language with OOP concepts to allow students to write code that is readable, maintainable, reusable, and overall more elegant.

This course includes Introduction to programming concepts (Procedural and Object Oriented Programming) with simple real-life projects thus, students will become familiar with project programming.

**Module 14**

| Code | Course/Module Title | ECTS | Semester |
|---|---|---|---|
| UOMCSSW2102 | **Software Engineering (1)** | 6 | 3 |
| **Class (hr/w)** | **Lect/Lab./Prac./Tutor** | **SSWL (hr/sem)** | **USWL (hr/w)** |
| 2 | 2+2 | 63 | 87 |
| **Description** | | | |

The Introduction in Software Engineering (1) course includes a detailed definition of software engineering , and give applications for all type of software and indicate the goals of the software engineering, and giving a brief explanation of the models, methods and tools of analysis phase

giving a brief explanation of the models, methods and tools of analysis phase in software engineering

**Module 15**

| Code | Course/Module Title | ECTS | Semester |
|------|---------------------|------|----------|
| UOMCSSW2103 | **Microprocessors and assembly language** | 6 | 3 |
| **Class (hr/w)** | **Lect/Lab./Prac./Tutor** | **SSWL (hr/sem)** | **USWL (hr/w)** |
| 2 | 2+2 | 63 | 87 |
| **Description** | | | |

Be familiar with the internal architecture of the 8086 microprocessor, and have to know how it works. This subject introduces 8086 microprocessors, which belongs to the Intel family of with an emphasis on the microprocessor based computer system, and the methods used to store data in a microprocessor-based system. Number systems and conversions are also included. Addressing mode and instruction code of 8086 microprocessors will be introduced.

Introduces the use of Visual C/C++ Express with the inline assembler and separate assembly language programming modules. It also explains how to configure Visual C++ express for use with assembly language applications. Once these basic are understood, provide applications using the Visual C++ Express with the inline assembler program include programming in the Windows environment.

**Module 16**

| Code | Course/Module Title | ECTS | Semester |
|------|---------------------|------|----------|
| UOMCSSW2104 | **Numerical Methods** | 2 | 3 |
| **Class (hr/w)** | **Lect/Lab./Prac./Tutor** | **SSWL (hr/sem)** | **USWL (hr/w)** |
| 2 | 1 | 33 | 17 |
| **Description** | | | |

Identifying the numerical methods used in solving nonlinear equations, completing and integrating, and addressing the concept of computer modeling and simulation processes through identifying the basic stages that are an essential and useful part of the mathematical modeling of many natural systems in physics (computational physics), astrophysics, chemistry and biology. And human systems in economics, psychology, social sciences and studies in the process of engineering new technology, enter into many industries and provide many experiences that need time and money. For an ongoing simulation that runs for several days. The scale of events simulated by computer simulations has far exceeded anything possible (or perhaps even imaginable).

**Module 17**

| Code | Course/Module Title | ECTS | Semester |
|------|---------------------|------|----------|
| UOMCSSW2105 | **Computability Theory** | 2 | 3 |
| **Class (hr/w)** | **Lect/Lab./Prac./Tutor** | **SSWL (hr/sem)** | **USWL (hr/w)** |
| 2 | 1 | 33 | 17 |
| **Description** | | | |

The Theory of Computability course aims to teach students to use algebraic laws in solving problems, laws of algebra of propositions, in addition to studying quantifiers, studying groups, theories, and different schemes, in addition to studying type of grammar, studying DFA and NDFA, state deletion algorithms, and many other topics

so that the student can solve problems related to theory. Calculation with ease and clarity.
This model aim to study Type of finite automata Define Regular expression (RE) Type of grammar Derivation trees for context free grammar Simplification of context free grammar Finite automata and their language Type of grammar The equivalence of DFA and NDFA Equivalence of FA and (RE) Kleene's theorem part 1 Kleene's theorem part 2 A general model of computation Turing machines that compute partial function Define Multiple Turing machines.

## Module 18

| Code | Course/Module Title | ECTS | Semester |
|---|---|---|---|
| UOMCSSW2106 | **Data Structure (1)** | 6 | 3 |
| **Class (hr/w)** | **Lect/Lab./Prac./Tutor** | **SSWL (hr/sem)** | **USWL (hr/w)** |
| 2 | 2+2 | 63 | 87 |
| **Description** | | | |

Data structures provide a way to efficiently manage large amounts of data, such as large databases and Internet indexing services. Usually, the efficiency of data structures is the key to designing efficient algorithms for some formal design methods and programming languages rely on data structures, rather than algorithms, thus becoming a major factor in software organization and design. Introduction to Data Structures, Algorithms, Stack, Queue, List, Trees, Sorting, Searching

## Module 19

| Code | Course/Module Title | ECTS | Semester |
|---|---|---|---|
| UOMCSSW2201 | **Software Engineering (2)** | 6 | 4 |
| **Class (hr/w)** | **Lect/Lab./Prac./Tutor** | **SSWL (hr/sem)** | **USWL (hr/w)** |
| 2 | 2+2 | 63 | 87 |
| **Description** | | | |

The Introduction in Software Engineering (2) course includes explaining its phases, and explain how modeling in each phase , and give applications for all phase Software Engineering giving a brief explanation of the models, methods and tools of all phases in software engineering

## Module 20

| Code | Course/Module Title | ECTS | Semester |
|---|---|---|---|
| UOMCSSW2202 | **Object Oriented Programming (2)** | 6 | 4 |
| **Class (hr/w)** | **Lect/Lab./Prac./Tutor** | **SSWL (hr/sem)** | **USWL (hr/w)** |
| 2 | 2+2 | 63 | 87 |
| **Description** | | | |

Teaching students:
- How to write readable, reusable, and modular Project.
- How to refactor their code periodically.

- How to test and maintain their code.

This course includes: Project's planning and requirements definition, class diagram designing, main phases definition, project development, and finally test and maintain their project.

## Module 21

| Code | Course/Module Title | ECTS | Semester |
|---|---|---|---|
| UOMCSSW2203 | **Database** | 4 | 4 |
| **Class (hr/w)** | **Lect/Lab./Prac./Tutor** | **SSWL (hr/sem)** | **USWL (hr/w)** |
| 2 | 2 | 33 | 67 |
| **Description** | | | |

The course reviews topics such as conceptual data modelling, relational data model, relational query languages, relational database design and transaction processing and current technologies such as semantic web, parallel and noSQL databases. It exposes the student to the fundamental concepts and techniques in database use and development as well provides a foundation for research in databases.

The course assumes prior exposure to databases, specifically to the relational data model and it builds new technologies on this foundation. In the first half of the course the relational data model, relational query languages, relational database design and conceptual data modeling are reviewed. It also bridges databases and knowledgebases which is the current trend. The course requires a term project in which the student implements a database application or explores a database issue. We will use PostgresSQL as the database platform for doing the assignments.

## Module 22

| Code | Course/Module Title | ECTS | Semester |
|---|---|---|---|
| UOMCSSW2204 | **Data Structure (2)** | 6 | 4 |
| **Class (hr/w)** | **Lect/Lab./Prac./Tutor** | **SSWL (hr/sem)** | **USWL (hr/w)** |
| 2 | 2+2 | 63 | 87 |
| **Description** | | | |

In computer science, a data structure is a specific way of storing and organizing data in a computer so that it can be used efficiently. Different types of data structures are suitable for different types of applications, and some are highly specialized for specific tasks. For example, (B-Tree) The binary tree is well suited for implementing databases, while compiler implementations usually use hash tables to look up identifiers. Data structures provide a way to efficiently manage large amounts of data, such as large databases and Internet indexing services. Usually, the efficiency of data structures is the key to designing efficient algorithms for some formal design methods and programming languages rely on data structures, rather than algorithms, thus becoming a major factor in software organization and design. Lists Stack Linked List Queue Linked List Recursion Trees Hashing.

## Module 23

| Code | Course/Module Title | ECTS | Semester |
|---|---|---|---|
| UOMCSSW2205 | **Systems Software** | 6 | 4 |
| **Class (hr/w)** | **Lect/Lab./Prac./Tutor** | **SSWL (hr/sem)** | **USWL (hr/w)** |

| 2 | 2+2 | 63 | 87 |

| Description |
|---|

This subject is basically focusing on the way of assemblers work by taking basic computer instructions and converts them into a pattern of bits that the computer's processor can use to perform its basic operations. It generates instructions by evaluating the mnemonics (symbols) in the operation field and find the value of symbols and literals to produce machine code. Be familiar with the internal architecture of the 8086 microprocessors, and have to know how does the 8086 works

## Module 24

| Code | Course/Module Title | ECTS | Semester |
|---|---|---|---|
| UOMCSSW2206 | **English Language(2)** | 2 | 4 |
| **Class (hr/w)** | **Lect/Lab./Prac./Tutor** | **SSWL (hr/sem)** | **USWL (hr/w)** |
| 2 | 1 | 33 | 17 |
| **Description** | | | |

Course Objectives: Learn English and practice conversation within the competence of IT. Basic details of the course The course consists of conversations in English with the specialization of computers (information technology), where the student receives all the information about the calculator and at the same time learns and trains on the correct pronunciation in this language as there are conversations between people about everything related to information technology). Working in the IT / IT Jobs and duties+ listen,Working in the IT / IT origination + listen Working in the IT / IT work rules + listen,Working in the IT / meeting + listen,Working in the IT / agenda + listen,Computer systems / computer hardware + listen,Computer systems / computer Software + listen,Computer systems / work with computers + listen IT Systems / system specifications + listen, IT Systems / GUI operations + listen,Data communications/ Email + listen.

## Module 25

| Code | Course/Module Title | ECTS | Semester |
|---|---|---|---|
| UOMCSSW3101 | **Artificial Intelligence** | 6 | 5 |
| **Class (hr/w)** | **Lect/Lab./Prac./Tutor** | **SSWL (hr/sem)** | **USWL (hr/w)** |
| 3 | 2+2 | 63 | 87 |
| **Description** | | | |

An artificial intelligence course is a specialized educational program that focuses on teaching the principles, theories, and applications of artificial intelligence (AI). It is designed to provide students with a comprehensive understanding of AI concepts, algorithms, and techniques, as well as hands-on experience in developing AI systems. The curriculum of an AI course typically covers a broad range of topics, starting with an introduction to AI and its historical development and fundamentals identifications of AI . Students also delve into topics like search methods like blind methods and Meta- heuristic methods. One of the central components of an AI course is Knowledge Representation typically focuses on the study of techniques and formalisms used to represent knowledge in computer systems. It explores ways to model and represent knowledge such as Propositional logic , First-order logic ,Inference rules and proof techniques and Semantic Networks, so that it can be effectively utilized by intelligent systems, such as expert systems, reasoning engines, and artificial intelligence applications. Another important aspect of AI courses is knowledge-based agent in artificial intelligence is an intelligent agent that utilizes knowledge representation and reasoning techniques to solve problems and make decisions, which

focuses on Inference and Reasoning and Knowledge Engineering. They may also explore advanced topics such as knowledge acquisition, knowledge elicitation techniques, knowledge verification, and knowledge refinement. Also Expert systems, also known as knowledge-based systems, are designed to simulate human expertise and provide intelligent solutions to specific problems or tasks. Additionally, an AI course often covers natural language processing (NLP), which involves teaching machines to understand, interpret, and generate human language. Students learn about techniques such as text classification, sentiment analysis, named entity recognition, and language modeling. They may also explore advanced topics like machine translation, question answering, and chatbot development. Throughout the course, students are likely to engage in hands-on programming assignments, where they implement AI algorithms using programming languages such as Matlab.

## Module 26

| Code | Course/Module Title | ECTS | Semester |
|------|---------------------|------|----------|
| UOMCSSW3102 | **Compilers Design** | 6 | 5 |
| **Class (hr/w)** | **Lect/Lab./Prac./Tutor** | **SSWL (hr/sem)** | **USWL (hr/w)** |
| 3 | 2+2 | 63 | 87 |
| **Description** | | | |

Programming languages are notations for describing computations to people and to machines. The world as we know it depends on programming languages, because all the software running on all the computers was written in some programming language. But, before a program can be run, it first must be translated into a form in which it can be executed by a computer .The software systems that do this translation are called compilers. A compiler is a program that reads a program written in one language, the source language, and translates it into an equivalent program in another language, the target language, as an important part of this translation process, the compiler reports to it's user the presence of errors in the source program.

## Module 27

| Code | Course/Module Title | ECTS | Semester |
|------|---------------------|------|----------|
| UOMCSSW3103 | **Software Project Management** | 4 | 5 |
| **Class (hr/w)** | **Lect/Lab./Prac./Tutor** | **SSWL (hr/sem)** | **USWL (hr/w)** |
| 3 | 2 | 33 | 67 |
| **Description** | | | |

The course aims to teach the student the basic concepts in software project management, which is one of the most important parts of software engineering. This includes knowing the basics of project scheduling, team building, risk management, choosing appropriate models for building and developing software projects, solving problems that hinder work on projects, and accurately completing them on time and within the limits of the allocated budgets. Software project management is an essential part of software engineering. Software projects need professional management in terms of planning an organizational budget and following models in the process of building and developing software, in addition to taking schedule constraints into account. Course content: Know the main tasks of software project managers, Software projects: types and nature, Determine the project work system, Estimating the needs, required materials, and cost of the project, Project time planning and implementation follow-up, Defining the importance of risk management and some of the risks that may arise in software projects, Understand the factors that influence personal motivation and what these might mean for software project managers, Understanding of key issues affecting teamwork, such as team composition, organization, and communication.
The expected outputs of this course are: Ability to plan software development projects, Determine the appropriate

models for building software projects according to the requirements required by the client, Listing and defining the project's internal and external resources, Estimated initial project cost and budget, Monitoring the implementation of the project plan and maintaining its completion, The ability to deal with and analyses risks, Leading and building the work team, he ability to adapt to the changes that may occur in the project during its implementation, Monitoring the project stages during implementation to ensure quality.

## Module 28

| Code | Course/Module Title | ECTS | Semester |
|---|---|---|---|
| UOMCSSW3104 | **Software Requirement** | 6 | 5 |
| **Class (hr/w)** | **Lect/Lab./Prac./Tutor** | **SSWL (hr/sem)** | **USWL (hr/w)** |
| 3 | 2+2 | 63 | 87 |
| **Description** | | | |

The curriculum aims to make the student familiar with the basic concepts and definitions associated with software requirements, and aims to provide the basic principles and methods used to achieve that allow ability and apply it correctly and at the required level according to the needs of the market.
Learning how to write good requirements, How to distinguish between functional and non-functional requirements, what are requirements engineering artifact modeling, what are methods and strategies used to elicit requirements.

## Module 29

| Code | Course/Module Title | ECTS | Semester |
|---|---|---|---|
| UOMCSSW3105 | **Computer Architecture** | 6 | 5 |
| **Class (hr/w)** | **Lect/Lab./Prac./Tutor** | **SSWL (hr/sem)** | **USWL (hr/w)** |
| 3 | 2+2 | 63 | 87 |
| **Description** | | | |

most computer systems used a single processor. On a singleprocessor system, there is one main CPUcapable of executing a general-purpose instruction set, including instructions from user processes. Almost all singleprocessor systems have other special-purpose processors as well. They may come in the form of device-specific processors, such as disk, keyboard, and graphics controllers; or, on mainframes, they may come in the form of more general-purpose processors, such as I/O processors that move data rapidly among the components of the system.

## Module 30

| Code | Course/Module Title | ECTS | Semester |
|---|---|---|---|
| UOMCSSW3106 | **English Language(3)** | 1 | 5 |
| **Class (hr/w)** | **Lect/Lab./Prac./Tutor** | **SSWL (hr/sem)** | **USWL (hr/w)** |
| 3 | 2 | 18 | 7 |
| **Description** | | | |

Course Objectives: Learn English and practice conversation within the competence of IT.Basic details of the course The course consists of conversations in English with the specialization of computers (information

technology), where the student receives all the information about the calculator and at the same time learns and trains on the correct pronunciation in this language as there are conversations between people about everything related to information technology). With the syllabus there is a tablet for conversations. Working in the IT / IT Jobs and duties + listen,Working in the IT / IT work rules + listen Working in the IT / meeting + listen,Working in the IT / agenda + listen,Working in the IT / multimedia hardware + listen,Working in the IT / operating systems + listen,Data communications/ internet browsing + listen,Data communications/ networks + listen Data communications/ mobile computing + listen,Data communications/ Email + listen.

## Module 31

| Code | Course/Module Title | ECTS | Semester |
|---|---|---|---|
| UOMCSSW3201 | **Intelligent techniques** | 6 | 6 |
| **Class (hr/w)** | **Lect/Lab./Prac./Tutor** | **SSWL (hr/sem)** | **USWL (hr/w)** |
| 3 | 2+2 | 63 | 87 |
| **Description** | | | |

A course on neural networks typically covers the theory, architecture, and practical applications of this fundamental component of artificial intelligence. Here is an outline of the key topics commonly included in a neural network course , Introduction to Neural Networks: This module provides an overview of neural networks, their historical background, and their role in AI. It covers the basic components of a neural network, including neurons, activation functions, and the feedforward process. It covers the architecture of perceptron , multilayer perceptron (MLPs), RBF,SOM and backpropagation algorithm for training. alse the course covered the genetic algorithm (GA) is a search and optimization technique inspired by the process of natural selection and evolution. It is used to find approximate solutions to optimization and search problems by mimicking the mechanics of biological evolution. Additionally course has fuzzy logic is a mathematical framework that deals with reasoning and decision-making in situations that involve uncertainty and imprecision. It is widely used in various fields, including control systems, artificial intelligence, and expert systems. If you're interested in studying fuzzy logic, here is a possible course outline that covers the fundamental concepts and applications.

## Module 32

| Code | Course/Module Title | ECTS | Semester |
|---|---|---|---|
| UOMCSSW3202 | **Software engineering tools** | 6 | 6 |
| **Class (hr/w)** | **Lect/Lab./Prac./Tutor** | **SSWL (hr/sem)** | **USWL (hr/w)** |
| 3 | 2+2 | 63 | 87 |
| **Description** | | | |

To help students understand the CASE tools in the software developments process. To introduce the many features of using the CASE tools throughout the project process. To enable students use different tools during the software development progress.
A CASE tool is a computer-based product supporting one  or more software engineering activities within a software development process. A CASE environment is a collection of CASE tools and other components together with an integration approach that supports most or all of the interactions that occur among the environment components, and between the users of the environment and the environment itself.
CASE tools are a class of software that automate many of the activities involved in various life cycle phases. For example, when establishing the functional requirements of a proposed application, prototyping tools can be used to develop graphic models of application screens to assist end users to visualize how an application will look after development. Subsequently, system designers can use automated design tools to transform the prototyped

functional requirements into detailed design documents. Programmers can then use automated code generators to convert the design documents into code. Automated tools can be used collectively, as mentioned, or individually. For example, prototyping tools could be used to define application requirements that get passed to design technicians who convert the requirements into detailed designs in a traditional manner using flowcharts and narrative documents, without the assistance of automated design software. Types of tools are: Business process engineering tools, Process modeling and management tools, Project planning tools, Risk analysis tools, etc.

## Module 33

| Code | Course/Module Title | ECTS | Semester |
|---|---|---|---|
| UOMCSSW3203 | **Software  fault tolerance** | 6 | 6 |
| **Class (hr/w)** | **Lect/Lab./Prac./Tutor** | **SSWL (hr/sem)** | **USWL (hr/w)** |
| 3 | 2+2 | 63 | 87 |
| **Description** | | | |

Software fault tolerance: is the property that enables a system to continue operating properly if it fails (or one or more faults within) some of its components. The descriptions of what the fault tolerance system should do— the services that it provides and the constraints on its operation. Through this course, students are learning what
- software fault tolerance is and distinguishes between error, fault, and failure.
-What are Dependable Fault-tolerant Systems?
- Failures, and Failure Modes.
- Fault tolerance technique used in the design and programming stage.
And other basic concepts that must be known to understand and apply Software Fault Tolerance.

## Module 34

| Code | Course/Module Title | ECTS | Semester |
|---|---|---|---|
| UOMCSSW3204 | **Operating Systems** | 6 | 6 |
| **Class (hr/w)** | **Lect/Lab./Prac./Tutor** | **SSWL (hr/sem)** | **USWL (hr/w)** |
| 3 | 2+2 | 63 | 87 |
| **Description** | | | |

An operating system (OS) is a collection of software that manages computer hardware resources and provides common services for computer programs. The operating system is a vital component of the system software in a computer system. Operating systems are an essential part of any computer system. Similarly, a course on operating systems is an essential part of any computer science education. This field is undergoing rapid change, as computers are now prevalent in virtually every arena of day-to-day life—from embedded devices in automobiles through the most sophisticated planning tools for governments and multinational firms. Yet the fundamental concepts remain fairly clear, and it is on these that we base this course.

## Module 35

| Code | Course/Module Title | ECTS | Semester |
|---|---|---|---|
| UOMCSSW3205 | **Network and Websites Engineering** | 6 | 6 |

| Class (hr/w) | Lect/Lab./Prac./Tutor | SSWL (hr/sem) | USWL (hr/w) |
|:---:|:---:|:---:|:---:|
| 3 | 2+2 | 63 | 87 |
| **Description** | | | |

Enabling students to be able to design and develop web pages, learn about the languages and means available to create web pages, deal with them, and program them, then learn about the most important standards that must be followed when designing web pages and applying engineering means to deal with available information and how it is modeled and structured in web pages to reach the main goal, which is Customer service, whether the purpose of displaying information on the page is scientific, industrial, commercial, etc., and all according to the need required and the availability of resources and tools for designing web tools.

**Module 36**

| Code | Course/Module Title | ECTS | Semester |
|:---:|:---:|:---:|:---:|
| UOMCSSW4101 | **Information Security** | 6 | 7 |
| **Class (hr/w)** | **Lect/Lab./Prac./Tutor** | **SSWL (hr/sem)** | **USWL (hr/w)** |
| 4 | 2+2 | 63 | 87 |
| **Description** | | | |

Giving an introduction to the attacks and their types that can occur during the transmission process between two parties and methods of data protection, either by traditional encryption methods, block encryption, encryption using the public key, how to choose the appropriate protection method, describing the attacks, and studying protection mechanisms, as required. The application of different encryption algorithms, with an explanation of the characteristics of each algorithm and its difference from other methods, and how to achieve the reliability and integrity of the data transmitted between two parties.

**Module 37**

| Code | Course/Module Title | ECTS | Semester |
|:---:|:---:|:---:|:---:|
| UOMCSSW4102 | **Software development techniques** | 4 | 7 |
| **Class (hr/w)** | **Lect/Lab./Prac./Tutor** | **SSWL (hr/sem)** | **USWL (hr/w)** |
| 4 | 2 | 33 | 67 |
| **Description** | | | |

to teach the student about old and modern software development techniques available. To understand and choose the appropriate technology from existing software development technologies for developing specific software depending on the size and nature of the project and its development team, and the rapid response to changes in the project. Compare between models from several aspects. To understand the benefits and disadvantages of each of these technologies.

**Module 38**

| Code | Course/Module Title | ECTS | Semester |
|:---:|:---:|:---:|:---:|

| UOMCSSW4103 | **Image and signal processing(1)** | 8 | 7 |
|---|---|---|---|
| **Class (hr/w)** | **Lect/Lab./Prac./Tutor** | **SSWL (hr/sem)** | **USWL (hr/w)** |
| 4 | 3+2 | 78 | 122 |
| **Description** | | | |

Defining about dealing with images and signal processing, in addition to using the techniques and methods for image processing and signal by converting the image from one domain to another, improving image and signal conversion methods, studying the properties of the transfers used in the study, as well as the process of enhancement the images and signals to improve their quality and luminance.

Where the subject dealt with the continuous and discrete Fourier transform in addition to all its characteristics (properties) that characterize this conversion, the fast and inverse Fourier transform and its qualities that this conversion possesses, the process of improving the image in the spatial domain and dealing with image data directly and improving the intensity of illumination of all kinds of images.

## Module 39

| Code | Course/Module Title | ECTS | Semester |
|---|---|---|---|
| UOMCSSW4104 | **Design of Real Time Systems** | 8 | 7 |
| **Class (hr/w)** | **Lect/Lab./Prac./Tutor** | **SSWL (hr/sem)** | **USWL (hr/w)** |
| 4 | 3+2 | 78 | 122 |
| **Description** | | | |

The main objectives are to introduce the basic concepts of real-time computing, illustrate the most significant results in the field, and provide the basic methodologies for designing predictable computing systems useful in supporting critical control applications. Real-time systems are computing systems that must react within precise time constraints to events in the environment. The correct behavior of these systems depends not only on the value of the computation but also on the time at which the results are produced.

First topic we dealt with is the concepts of many definitions of real time system such as predictability, deadline and misses deadline categories (hard, firm, soft) The topics discussed the limits of current real-time systems as multitasking, priority-based scheduling, ability to quickly respond to external interrupts, basic mechanisms for process communication and synchronization, small kernel and fast context switch and absence of a real-time clock as an internal time reference. Another topic, broached to absence of a real-time clock as an internal time reference that must have to support critical applications such as timeliness, predictability, efficiency, robustness, fault tolerance and maintainability. Also we broached with another subject of how to deal with many algorithms that lead to time predictability to keep the deadline of critical tasks that must end on time specified that works on uniprocessor such as rate-monotonic scheduling algorithm, earliest deadline first scheduling algorithm, awaiting service, deferred server, also the topic was covered the priority inversion phenomenon where it processed with priority inheritance and priority ceiling. Then, several assignment algorithms were discussed that run tasks on many CPUs such as next-fit algorithm, bin packing algorithm and myopic algorithm.

## Module 40

| Code | Course/Module Title | ECTS | Semester |
|---|---|---|---|
| UOMCSSW4105 | **Software Support** | 2 | 7 |
| **Class (hr/w)** | **Lect/Lab./Prac./Tutor** | **SSWL (hr/sem)** | **USWL (hr/w)** |
| 4 | 1 | 33 | 17 |

| Description |
|---|
| The course aims to teach the student that the program development process continues all the time even after the program is delivered to the customer, as a result of errors appearing in it and being corrected, or to adapt the program to work in a new environment, or as a result of the customer's request for new functions and features of the program.<br>The course introduces students to everything related to maintenance, re-engineering and reverse engineering. |

**Module 41**

| Code | Course/Module Title | ECTS | Semester |
|---|---|---|---|
| UOMCSSW4106 | **English Language(4)** | 2 | 7 |
| **Class (hr/w)** | **Lect/Lab./Prac./Tutor** | **SSWL (hr/sem)** | **USWL (hr/w)** |
| 4 | 1 | 33 | 17 |
| **Description** | | | |

| Description |
|---|
| Course Objectives: Learn English and practice conversation within the competence of IT2 Basic details of the course The course consists of conversations in English with the specialization of computers (information technology), where the student receives all the information about the calculator and at the same time learns and trains on the correct pronunciation in this language as there are conversations between people about everything related to information technology). With the syllabus there is a tablet for conversations. Working in the IT2 / IT2 CH 5 Choice/ compare products+listen,Working in the IT2 / IT2 CH 5 Choice / discuss IT costs +listen,Working in the IT2 / IT2 CH 5 Choice/ research products products+listen,Working in the IT2 / IT2 CH 5 Choice/ recommend products+listen. Working in the IT2 / IT2 CH 6 interactions / describe trends+listen,Working in the IT2 / IT2 CH 6 interactions / describe benefits of video+listen,Working in the IT2 / IT2 CH 6 interactions / describe trends+listen. |

**Module 42**

| Code | Course/Module Title | ECTS | Semester |
|------|---------------------|------|----------|
| UOMCSSW4201 | **Computer Network Security** | 6 | 8 |
| **Class (hr/w)** | **Lect/Lab./Prac./Tutor** | **SSWL (hr/sem)** | **USWL (hr/w)** |
| 4 | 2+2 | 63 | 87 |
| **Description** | | | |

Teaching students to protect data with different concealment methods or how to choose the appropriate concealment method, giving a description of intruders, studying protection mechanisms from intrusion, methods for choosing passwords, and how to achieve reliability and integrity of data when sending, in addition to giving computer viruses their types and methods of protection from them.
In this course, methods of concealment are taught in the spatial domain and frequency domain, types of intrusion, methods for choosing passwords, in addition to achieving data integrity when transmitting, as well as viruses, their types, and methods of protection against them.

**Module 43**

| Code | Course/Module Title | ECTS | Semester |
|------|---------------------|------|----------|
| UOMCSSW4202 | **Software Quality Assurance** | 2 | 8 |
| **Class (hr/w)** | **Lect/Lab./Prac./Tutor** | **SSWL (hr/sem)** | **USWL (hr/w)** |
| 4 | 1 | 33 | 17 |
| **Description** | | | |

Teaching students how to test software, methods and types of software testing strategies, and achieving software testing quality assurance. Teaching the student how to test software to find and correct errors in the developed software. To understand and choose from existing software testing methods and strategies. To achieve well-developed software quality assurance for software. To manage the software quality assurance process and periodic reviews.

**Module 44**

| Code | Course/Module Title | ECTS | Semester |
|------|---------------------|------|----------|
| UOMCSSW4203 | **Image and signal processing(2)** | 6 | 8 |
| **Class (hr/w)** | **Lect/Lab./Prac./Tutor** | **SSWL (hr/sem)** | **USWL (hr/w)** |
| 4 | 2+2 | 63 | 87 |
| **Description** | | | |

Getting acquainted with dealing with images and signals, in addition to the use of methods and techniques for image and signal processing to enhance images and signals to improve their quality and illumination, in addition to identifying both types of compression methods and quality standards for the method used by pressing, adding methods for defining edges and other ways of improving and preparing data for the image and signal.

The course dealt with the process of enhancing the image in the spatial domain, dealing with image data directly, improving the intensity of illumination of all types of images, as well as improving the image in the frequency domain, in addition to methods for determining image edges and cutting images, as well as data clustering. In addition to studying the installation of filters for signal processing.

**Module 45**

| Code | Course/Module Title | ECTS | Semester |
|---|---|---|---|
| UOMCSSW4204 | **Elective** | 4 | 8 |
| **Class (hr/w)** | **Lect/Lab./Prac./Tutor** | **SSWL (hr/sem)** | **USWL (hr/w)** |
| 4 | 2 | 33 | 67 |
| **Description** | | | |

**Elective : Mobile Programming, Social Media Networking, Open Source Software, software reliability**

**Mobile Programming**
  Introduction to smart phones, Systems model of mobile platform diffusion, Smartphones types
  Mobile phone main components, Smartphone hardware architecture, Smartphone software's
  Mobile Operating System, System Architecture, iPhone/Ios, ANDROID OPERATING SYSTEM (OS)
  Android Interface, Utilities Android OS, Application framework, INTERACTION MODELS
  General plan for integrated Android application, Main characteristics of a distributed system


 **Social Media Networking**
Study and application of social media communication and strategy.  Including social media platforms and user devices, message distribution, and personal and professional online, social environments. Indicative topics to be covered are likely to include: Introduction to digital and social media marketing, Online marketplace analysis, Digital marketing strategy, The impact of digital media and technology on the marketing mix, Relationship marketing using digital platforms,Delivering the online customer experience, Campaign planning for digital media, Marketing communications using digital media platforms.


 **Open Source Software**
Introduction to Open Source Software is for anyone who uses a computer and is tired of paying licensing fees for software, but concerned about downloading nasties from the Internet. In a friendly, step-by-step environment, the course takes you through essential concepts like open source, closed source, copyright and types of licenses, …. etc., this course also will enabled you to know the type of software you choose shareware or freeware. At the end of the course, you'll be equipped to do your day-to-day computing tasks without ever needing to pay for software again.

**Software reliability**
Software reliability aims to define Reliability in general and to address Hardware and Software Reliability, Software Reliability Engineering (SRE) with a Software reliability analysis containing component configurations (series and parallel). In addition, the basic reliability metrics along with some of the software metrics related to Reliability are presented. Moreover, Software Reliability tools are discussed with some details, giving as an example some of the available and recognized tools that can be used to assess software reliability. Furthermore, the four well-known types of Failure Mode and Effect Analysis (FMEA) are introduced and explained, together with the process of conducting an effective FMEA. Along with the previously mentioned subjects of software reliability, Software Reliability Models and Software Reliability Improvement Techniques are presented.

**Module 46**

| Code | Course/Module Title | ECTS | Semester |
|---|---|---|---|
| UOMCSSW4205 | **Internet of Things** | 6 | 8 |
| **Class (hr/w)** | **Lect/Lab./Prac./Tutor** | **SSWL (hr/sem)** | **USWL (hr/w)** |
| 4 | 2+2 | 63 | 87 |
| **Description** | | | |
| Learn what is the IoT and how it can be used in reality. Understand how the IoT is bridging the gap between operational and information technology systems. Also, the security concerns that must be considered when implementing IoT solutions. To provide an overview of IoT concepts, principles, and applications. To understand the layered architecture of IoT, including sensors and actuators, connectivity, cloud platforms, and applications. Learn different types of sensors used in IoT devices. To learn how explore the different connectivity options for IoT devices. To gain knowledge of data management techniques for IoT. To learn about real-time analytics, data visualization, and extracting insights from IoT-generated data. To understand the security challenges and vulnerabilities associated with IoT systems. To Study real-world examples and case studies of IoT applications across various industries. Learn to develop and program IoT devices using platforms . To explore the integration of IoT devices with cloud platforms. | | | |

**Module 47**

| Code | Course/Module Title | ECTS | Semester |
|---|---|---|---|
| UOMCSSW4206 | **Project** | 6 | 8 |
| **Class (hr/w)** | **Lect/Lab./Prac./Tutor** | **SSWL (hr/sem)** | **USWL (hr/w)** |
| 4 | 3 | 33 | 117 |
| **Description** | | | |
| This section includes a description of the module, 100-150 words | | | |

# Contact

Program Manager:

Naktal Moaid Edan   | PhD in Computer Science   |   Assistant Prof.

Email: naktal.edan@uomosul.edu.iq

Mobile no.: 07721212333

Program Coordinator:

Yasir Mohammed khazaal    | MSc. in software   |   Assistant Lecturer

Email: yasirmk@uomosul.edu.iq

Mobile no.: 07740889760