# المحاضرة الثانية

# Classes in Python

# Q/ Python Code for Bank Account System Using Classes

```
class BankAccount:
    def__ init__(self, account_holder, balance=0):
        self.account_holder = account_holder
        self.balance = balance

    def deposit(self, amount):
         if amount > 0:
            self.balance += amount
            print(f"تم إيداع {amount}$. الرصيد الجديد: {self.balance}$")
        else:
            print("خطأ: لا يمكن إيداع مبلغ أقل من أو يساوي الصفر.")
```

# Q/ Python Code for Bank Account System Using Classes

```
def withdraw(self, amount):
    if 0 < amount <= self.balance:
        self.balance -= amount
        print(f"تم سحب {amount}$. الرصيد المتبقي: {self.balance}$")
    else:
        print("خطأ: المبلغ المطلوب للسحب غير متاح أو غير صالح.")
```

# Q/ Python Code for Bank Account System Using Classes

```
def display_balance(self):
    print(f“ name: {self.account_holder}, current balance: {self.balance}$")
account1 = BankAccount (“Ali”, 1000)
account1.display_balance()
account1.deposit(500
account1.withdraw(300)
account1.withdraw(2000)
account1.display_balance()
```

# Q/ Graph Implementation Using Classes in Python

```
class Graph:
    def __init__(self):
        self.graph = {}
    def add_node(self, node):
        if node not in self.graph:
            self.graph[node] = []
```

## Q/ Graph Implementation Using Classes in Python

```
def add_edge(self, node1, node2, bidirectional=True):
        if node1 not in self.graph:
            self.add_node(node1)
        if node2 not in self.graph:
            self.add_node(node2)
        self.graph[node1].append(node2)
        if bidirectional:
            self.graph[node2].append(node1)
```

# Q/ Graph Implementation Using Classes in Python

```
def display(self):
    for node, neighbors in self.graph.items():
        print(f"{node} --> {neighbors}")
g = Graph()
g.add_edge("A", "B")
g.add_edge("A", "C")
g.add_edge("B", "D", bidirectional=False)
g.add_edge("C", "D")
g.display()
```

**# Implementation**

```
A--> ['B', 'C']
B --> ['A', 'D']
C --> ['A', 'D']
D --> ['C']
```

# Thank you