Experiment -3

Theory:-

- Class is a blueprint that help create objects
- A class contains:-
  - Instructions on how to create object
  - Define it's properties and behaviours
- Inheritance allows class to inherit attributes and methods from another class.
- It establishes a hierarchy among classes.
- Subclass can use methods of it's superclass.
- This helps code reusability.

Example

- Define a class called 'vehicle'
  → Give it attributes like 'colour', 'wheel'.
- Define other classes like car, bike which inherit methods from vehicles and then add their own unique features.

Conclusion:-

Class form the backbone of object oriented programming which helps develop robust code. It also helps in organisation as you can keep similar functionality code together which enhances readability.

Exp3.py - C:\asg\Python\Exp3.py (3.11.2)

File Edit Format Run Options Window Help

```
print("Sharan Shetty 60004220224")
class Employee:
    def __init__(self, name, emp_id):
        self.name = name
        self.emp_id = emp_id

    def display_info(self):
        print(f"Employee Name: {self.name} \nEmployee ID: {self.emp_id}")


class Developer(Employee):
    def __init__(self, name, emp_id):
        super().__init__(name, emp_id)
        self.role = "Developer"


class Tester(Employee):
    def __init__(self, name, emp_id):
        super().__init__(name, emp_id)
        self.role = "Tester"


class Manager(Employee):
    def __init__(self, name, emp_id):
        super().__init__(name, emp_id)
        self.role = 'Manager'
        self.managed_developers = []
        self.managed_testers = []

    def add_employee(self, employee):
        if employee.role == 'Developer':
            self.managed_developers.append(employee)
            print(f"Developer {employee.name} is added to Manager {self.name}'s team.")

        elif employee.role == 'Tester':
            self.managed_testers.append(employee)
            print(f"Tester {employee.name} is added to Manager {self.name}'s team.")

        else:
            print(f"Employee {employee.name} is neither a Developer nor Tester and cannot be added to the team.")

    def remove_employee(self, ID, role):
        if role == 'Developer':
            for employee in self.managed_developers:
                if ID == employee.emp_id:
                    self.managed_developers.remove(employee)
                    print(f"{employee.name} removed from manager's team.")
```

Ln: 1 Col: 34

Exp3.py - C:\asg\Python\Exp3.py (3.11.2)

File Edit Format Run Options Window Help

```
                    return
            print(f"Tester with {ID} ID is not managed by this manager.")

    def display_managed_employees(self):
        if self.managed_developers:
            print(f"Developers managed by {self.name}:")
            for employee in self.managed_developers:
                employee.display_info()
        else:
            print(f"{self.name} does not manage any developers at the moment.")

        if self.managed_testers:
            print(f"Testers managed by {self.name}:")
            for employee in self.managed_testers:
                employee.display_info()
        else:
            print(f"{self.name} does not manage any testers at the moment.")


options = '''
1. Add Manager   2. Choose Manager   3. Add Developer   4. Add Tester
5. Remove Developer   6. Remove Tester   7. Print all Employees under Current Manager
-1. QUIT

Enter Choice:
'''
managers = []
choice = 1

while choice != -1:
    if choice == 1:
        m_name = input("Enter Name of Manager:")
        m_id = int(input("Enter Employee ID of Manager:"))
        new_manager = Manager(m_name, m_id)
        managers.append(new_manager)
        manager = new_manager

    elif choice == 2:
        print("The managers are:")
        for m in managers:
            print(m.name)

        m_chosen_id = int(input("Enter Manager ID Chosen:"))
        for m in managers:
            if m.emp_id == m_chosen_id:
                manager = m
```

Ln: 1 Col: 34

Exp3.py - C:\asg\Python\Exp3.py (3.11.2)

File Edit Format Run Options Window Help

```
choice = 1

while choice != -1:
    if choice == 1:
        m_name = input("Enter Name of Manager:")
        m_id = int(input("Enter Employee ID of Manager:"))
        new_manager = Manager(m_name, m_id)
        managers.append(new_manager)
        manager = new_manager

    elif choice == 2:
        print("The managers are:")
        for m in managers:
            print(m.name)

        m_chosen_id = int(input("Enter Manager ID Chosen:"))
        for m in managers:
            if m.emp_id == m_chosen_id:
                manager = m
                break

    elif choice == 3:
        dev_name = input("Enter Name of Developer:")
        dev_id = int(input("Enter Employee ID of Developer:"))
        new_developer = Developer(dev_name, dev_id)
        manager.add_employee(new_developer)

    elif choice == 4:
        tes_name = input("Enter Name of Tester:")
        tes_id = int(input("Enter Employee ID of Tester:"))
        new_tester = Tester(tes_name, tes_id)
        manager.add_employee(new_tester)

    elif choice == 5:
        rem_id = int(input("Enter Employee ID of Developer to be removed:"))
        manager.remove_employee(rem_id, 'Developer')

    elif choice == 6:
        rem_id = int(input("Enter Employee ID of Tester to be removed:"))
        manager.remove_employee(rem_id, 'Tester')

    elif choice == 7:
        manager.display_managed_employees()

    choice = int(input(options))
```

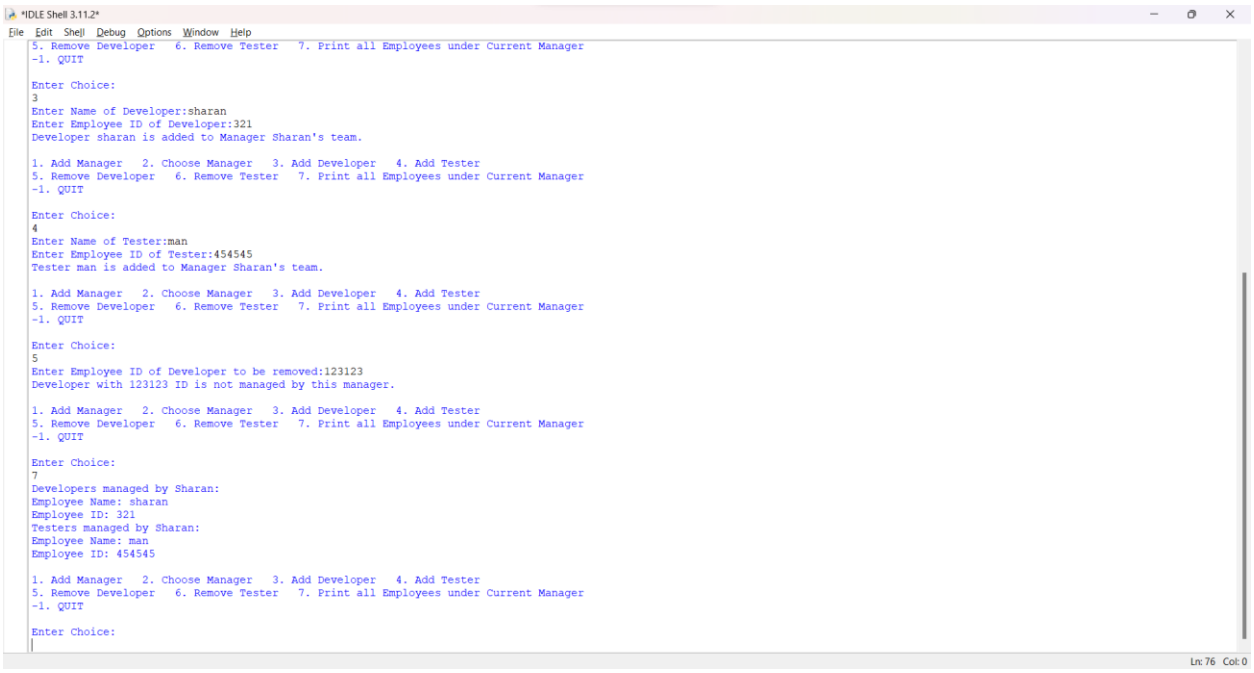
Ln: 113 Col: 0

IDLE Shell 3.11.2

File Edit Shell Debug Options Window Help

```
Python 3.11.2 (tags/v3.11.2:878ead1, Feb  7 2023, 16:38:35) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:\asg\Python\Exp3.py ==========
Sharan Shetty 60004220224
Enter Name of Manager:Sharan
Enter Employee ID of Manager:123123

1. Add Manager   2. Choose Manager   3. Add Developer   4. Add Tester
5. Remove Developer   6. Remove Tester   7. Print all Employees under Current Manager
-1. QUIT

Enter Choice:
1
Enter Name of Manager:rahul
Enter Employee ID of Manager:212121

1. Add Manager   2. Choose Manager   3. Add Developer   4. Add Tester
5. Remove Developer   6. Remove Tester   7. Print all Employees under Current Manager
-1. QUIT

Enter Choice:
2
The managers are:
Sharan
rahul
Enter Manager ID Chosen:123123

1. Add Manager   2. Choose Manager   3. Add Developer   4. Add Tester
5. Remove Developer   6. Remove Tester   7. Print all Employees under Current Manager
-1. QUIT

Enter Choice:
3
Enter Name of Developer:sharan
Enter Employee ID of Developer:321
Developer sharan is added to Manager Sharan's team.

1. Add Manager   2. Choose Manager   3. Add Developer   4. Add Tester
5. Remove Developer   6. Remove Tester   7. Print all Employees under Current Manager
-1. QUIT

Enter Choice:
4
Enter Name of Tester:man
Enter Employee ID of Tester:454545
Tester man is added to Manager Sharan's team.
```

Ln: 76 Col: 0


IDLE Shell 3.11.2
File Edit Shell Debug Options Window Help
5. Remove Developer 6. Remove Tester 7. Print all Employees under Current Manager
-1. QUIT
Enter Choice:
3
Enter Name of Developer:sharan
Enter Employee ID of Developer:321
Developer sharan is added to Manager Sharan's team.
1. Add Manager 2. Choose Manager 3. Add Developer 4. Add Tester
5. Remove Developer 6. Remove Tester 7. Print all Employees under Current Manager
-1. QUIT
Enter Choice:
4
Enter Name of Tester:man
Enter Employee ID of Tester:454545
Tester man is added to Manager Sharan's team.
1. Add Manager 2. Choose Manager 3. Add Developer 4. Add Tester
5. Remove Developer 6. Remove Tester 7. Print all Employees under Current Manager
-1. QUIT
Enter Choice:
5
Enter Employee ID of Developer to be removed:123123
Developer with 123123 ID is not managed by this manager.
1. Add Manager 2. Choose Manager 3. Add Developer 4. Add Tester
5. Remove Developer 6. Remove Tester 7. Print all Employees under Current Manager
-1. QUIT
Enter Choice:
7
Developers managed by Sharan:
Employee Name: sharan
Employee ID: 321
Testers managed by Sharan:
Employee Name: man
Employee ID: 454545
1. Add Manager 2. Choose Manager 3. Add Developer 4. Add Tester
5. Remove Developer 6. Remove Tester 7. Print all Employees under Current Manager
-1. QUIT
Enter Choice:
Ln: 76 Col: 0