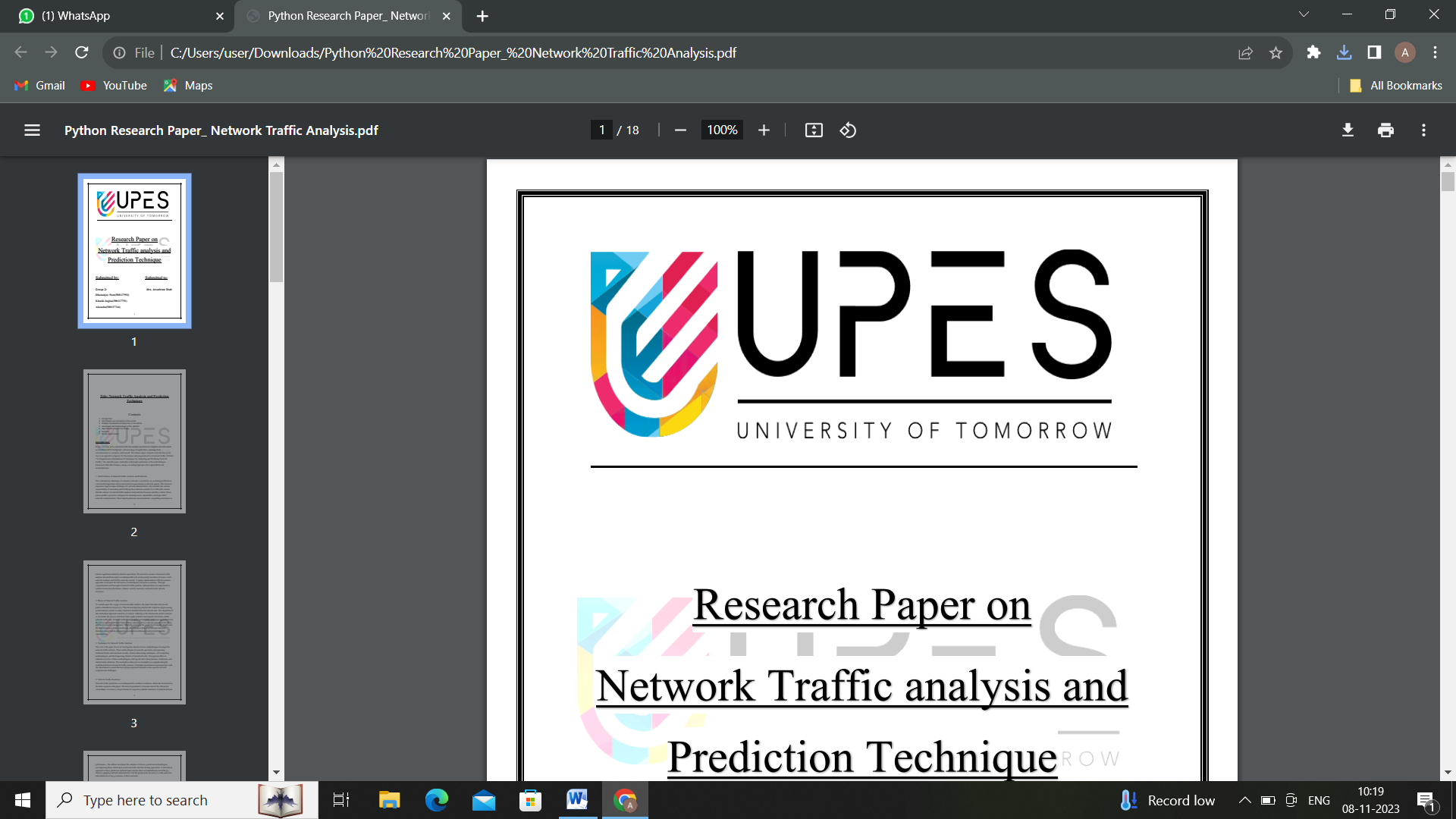
****

**ASSIGNMENT-6**

**SUBMITTED TO: SUBMITTED BY:**

**Saurabh Jain Akansha**

**500117744**

**MCA batch-1**

Que-1)

**Algorithm:**

**Step-1:** Define an interface "A" with methods "meth1" and "meth2".

**Step-2**: Create a class "MyClass" that implements interface "A".

**Step-3:** Implement methods "meth1" and "meth2" in "MyClass".

**Step-4:** In the main program, create an object of "MyClass" and call its methods "meth1" and "meth2".

Code:

interface A {

    void *methh1*();

    void *methh2*();

}

class MineClass implements A {

    @Override

    public void *methh1*() {

*System*.*out*.*println*("This is inside method 1 of java  MineClass");

    }

    @Override

    public void *methh2*() {

*System*.*out*.*println*("This is inside method 1 of java  MineClass");

    }

}

public class Main {

    public static void *main*(String[] *args*) {

        MineClass *obj* = **new** *MineClass*();

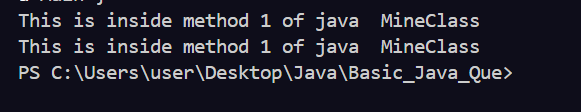
*obj*.*methh1*();

*obj*.*methh2*();

    }

}

Output:



**Que-2)**

**Algorithm:**

1. Define an interface "Machine" with methods "powerOn" and "powerOff".

2. Define an interface "Vehicle" extending "Machine" with methods "moveForward" and "moveBackward".

3. Define an interface "ElectricVehicle" extending "Vehicle" with a method "chargeBattery".

4. Implement a class "ElectricCar" implementing "ElectricVehicle" and providing concrete implementations for all methods.

5. In the main program:

- Create an object of "ElectricCar" named "myElectricCar".

- Call the method "powerOn" on "myElectricCar" to start the electric car.

- Call the methods "moveForward" and "moveBackward" to simulate the movement of the electric car.

- Call the method "chargeBattery" to charge the battery of the electric car.

- Call the method "powerOff" to shut down the electric car.

**Code:**

interface Machine {

    void *powerOn*();

    void *powerOff*();

}

interface Vehicle extends Machine {

    void *moveForward*();

    void *moveBackward*();

}

interface ElectricVehicle extends Vehicle {

    void *chargeBattery*();

}

class ElectricCar implements ElectricVehicle {

    @Override

    public void *powerOn*() {

*System*.*out*.*println*("Electric car starting...");

    }

    @Override

    public void *powerOff*() {

*System*.*out*.*println*("Electric car shutting down...");

    }

    @Override

    public void *moveForward*() {

*System*.*out*.*println*("Electric car moving forward...");

    }

    @Override

    public void *moveBackward*() {

*System*.*out*.*println*("Electric car moving backward...");

    }

    @Override

    public void *chargeBattery*() {

*System*.*out*.*println*("Electric car charging battery...");

    }

}

public class Main {

    public static void *main*(String[] *args*) {

        ElectricCar *myElectricCar* = **new** *ElectricCar*();

*myElectricCar*.*powerOn*();

*myElectricCar*.*moveForward*();

*myElectricCar*.*moveBackward*();

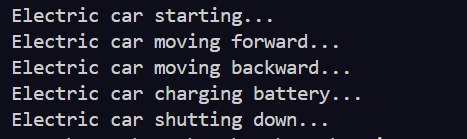
*myElectricCar*.*chargeBattery*();

*myElectricCar*.*powerOff*();

    }

}

Output:



Que-3)

Algorithm:

1. Define an interface named "Operations" with methods "performDivision" and "performModule".

2. Create a class named "Calculator" that implements "Operations" and provides concrete implementations for its methods.

3. In the main program:

• Create an object of "Calculator" named "calc".

• Call the methods "performDivision" and "performModule" on the "calc" object to demonstrate the functionality of the calculator.

Code:

interface Operations {

    void *performDivision*();

    void *performModule*();

}

class Calculator implements Operations {

    @Override

    public void *performDivision*() {

*System*.*out*.*println*("Inside performDivision method in Calculator");

    }

    @Override

    public void *performModule*() {

*System*.*out*.*println*("Inside performModule method in Calculator");

    }

}

public class Main {

    public static void *main*(String[] *args*) {

        Calculator *calc* = **new** *Calculator*();

*calc*.*performDivision*();

*calc*.*performModule*();

    }

}

Output:



Que-4)

**Algorithm:**

* Define an interface named "Calculation" with a method "performSquare".
* Create a class named "Calculator" that implements the "Calculation" interface and provides a concrete implementation for the "performSquare" method.
* Create a new class named "CalculateSquare" to demonstrate the use of the "Calculator" class.
* In the "CalculateSquare" class:
* Create an object of the "Calculator" class.
* Use the "performSquare" method from the "Calculation" interface via the "Calculator" object.
* Print the result of squaring a number to demonstrate the functionality.

**Code:**

interface Calculation {

    int *performSquare*(int *num*);

}

class Calculator implements Calculation {

    @Override

    public int *performSquare*(int *num*) {

**return** num \* num;

    }

}

public class Main {

    public static void *main*(String[] *args*) {

        Calculator *calculator* = **new** *Calculator*();

        int *result* = *calculator*.*performSquare*(8);

*System*.*out*.*println*("Square of 8 is: " + result);

    }

}

Output:

