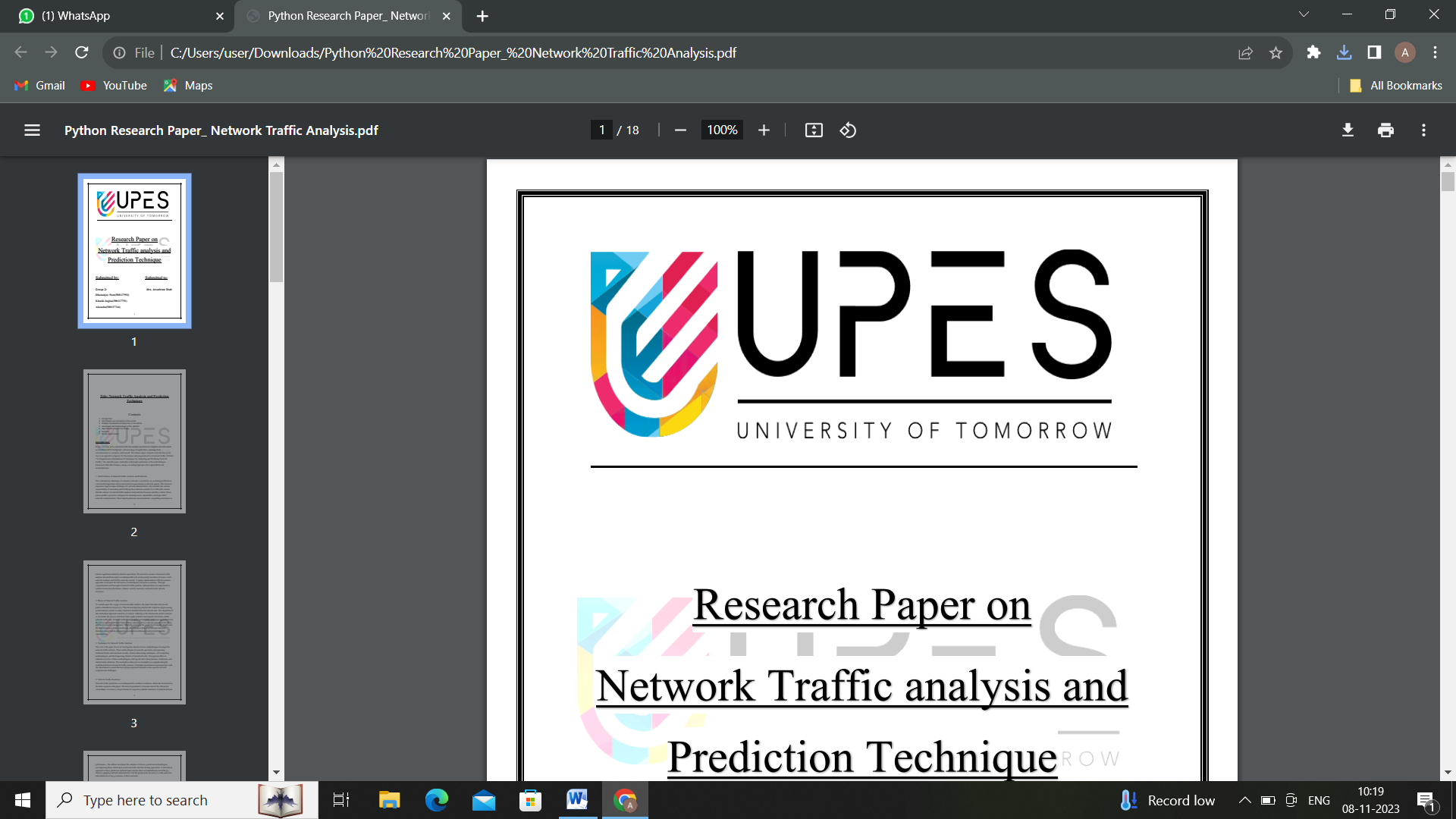
****

**ASSIGNMENT-5**

**SUBMITTED TO: SUBMITTED BY:**

**Saurabh Jain Akansha**

**500117744**

**MCA batch-1**

**Ans-1)**

**Algorithm of code:**

**Step-1:** First of all create a class named “Access\_Member”, in this class add a private member like SAP.

**Step-2:** Then, create a class named Access\_Member2 which extends the previous class.

**Step-3:** Then we have to call the extended class and try to access the SAP attribute that is private. Now we will see an error.

class Access\_Member

{

private int SAP= 500117744;

}

class Access\_Member2 extends Access\_Member

{

String name= "Akansha";

int Roll= 30;

}

class Main{

public static void main(String args[])

{

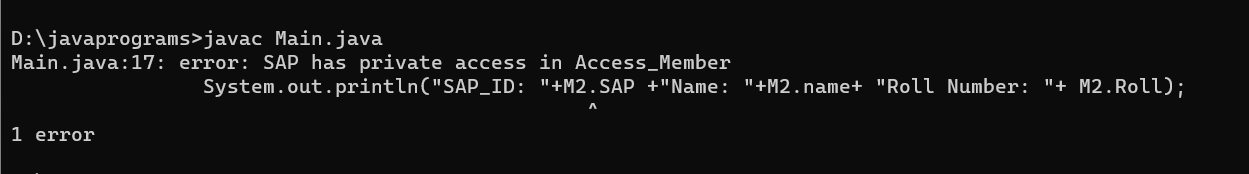
Access\_Member2 M2= new Access\_Member2();

System.out.println("SAP\_ID: "+M2.SAP +"Name: "+M2.name+ "Roll Number: "+ M2.Roll);

}

}

**Output:**

****

**Ans-2)**

**Algorithm of the code:**

**Step-1:** First step would be to create a class “Player” Which will be further extended by the classes named as CricketPlayer, FootballPlayer and HockeyPlayer.

**Step-2:** Now we have to make objects of the three classes in the main class in which we will call the function of the three classes name as details.

**Step-3:** We can then see that the name and jersy number is accessed by all classed that are derived.

**Code:**

class Player{

String name= "Rahul";

int Jersy\_no= 5;

}

class CricketPlayer extends Player

{

void details()

{

System.out.println(name+ " Plays Cricket with jersy number "+ Jersy\_no);

}

}

class FootballPlayer extends Player

{

void details()

{

System.out.println(name+ " Plays Football with jersy number "+ Jersy\_no);

}

}

class HockeyPlayer extends Player

{

void details()

{

System.out.println(name+ " Plays Hockey with jersy number "+ Jersy\_no);

}

}

public class PlayerData

{

public static void main(String[] args)

{

CricketPlayer Cricket= new CricketPlayer();

FootballPlayer Football= new FootballPlayer();

HockeyPlayer Hockey= new HockeyPlayer();

Cricket.details();

Football.details();

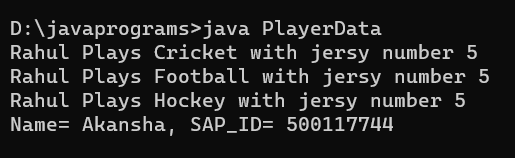
Hockey.details();

System.out.println("Name= Akansha, SAP\_ID= 500117744");

}

}

**Output:**



**Ans-3)**

**Algorithm of the code:**

**Step-1:** Make a class “Worker” which have the name of worker and salary of worker as two attributes.

**Step-2:** Then, we will make constructor which will inilialise the above attributes.

**Step-3:** Then we will make a common method named as ComPay which will be same by name but different by functionality for both the workers.

**Step-4:** At last we will call the method by making objects of classes and we will print their salary.

**Code:**

//Worker class

abstract class Worker {

protected String WorkerName;

protected double WorkerSalary;

public Worker(String WorkerName, double WorkerSalary) {

this.WorkerName = WorkerName;

this.WorkerSalary = WorkerSalary;

}

public abstract double comPay(int hours);

}

class SalariedWorker extends Worker {

public SalariedWorker(String WorkerName, double WorkerSalary) {

super(WorkerName, WorkerSalary);

}

//overrides

public double comPay(int hours) {

return WorkerSalary \* 40;

}

}

//DailyWorker class

class DailyWorker extends Worker {

public DailyWorker(String WorkerName, double WorkerSalary) {

super(WorkerName, WorkerSalary);

}

//overrides

public double comPay(int days) {

return WorkerSalary \* days;

}

}

public class WorkerData {

public static void main(String[] args) {

DailyWorker daily = new DailyWorker("Amit", 100);

SalariedWorker salaried = new SalariedWorker("Sumit", 25);

System.out.println("Daily Worker pay " + daily.comPay(6));

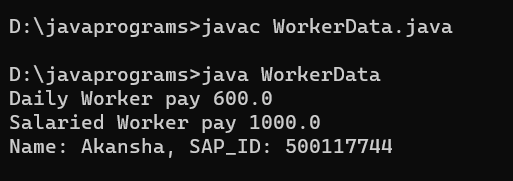
System.out.println("Salaried Worker pay " + salaried.comPay(45));

System.out.println("Name: Akansha, SAP\_ID: 500117744");

}

}

Output:



**Ans-4)**

**Algorithm for code:**

**Step-1:** First of all we will make a class named Staff which contain basic information like Staff income and Staff SAP.

**Step-2:** In order to get the name and income of the Staff we will create two methods.

**Step-3:** After this we will extends the Staff class and make another class called as Senior. Senior class entends Staff.

**Step-4:** Next we will create two different objects in order to access our classes.

**Step-5:** At last we will print their new income after increasing it by some extend.

**Code:**

class Staff {

private String memb;

private String SAP;

private double income;

public Staff() {}

public Staff(String memb, String SAP, double income) {

this.memb = memb;

this.SAP = SAP;

this.income = income;

}

public String InputName() {

return memb;

}

public double InputSalary() {

return income;

}

public void IncSalary(double percent) {

income += income + income \* percent / 100;

}

}

class Senior extends Staff {

public Senior(String memb, String SAP, double income) {

super(memb, SAP, income);

}

}

public class Office {

public static void main(String[] args) {

Staff em = new Staff("Aradhya", "E89", 77889);

em.IncSalary(11);

System.out.println(em.InputName() + " Income is: " + em.InputSalary());

Senior Se = new Senior("Akansha", "U997", 1200000);

Se.IncSalary(25);

System.out.println(Se.InputName() + " Income is: " + Se.InputSalary());

System.out.println("Name is Akansha and SAP\_ID is 500117744");

}

}

**Output:**

