**UNIVERSITY OF RWANDA**

**CBE GIKONDO CAMPUS**

**YEAR 2**

**DEPARTMENT OF BUSINESS AND INFORMATION TECHNOLOGY**

**NAMES: MUTESAYIRE CARINE**

**REG NUMBER: 221020200**

**QUIZ OF JAVA**

**TOPIC 1**

### FIRST EXPERIMENT ON JAVA CONTROL STATEMENT:  if-else statement

**SOURCE CODES ON JAVA CONTROL STATEMENT ON IF ELSE STATEMENT ARE THE FOLLOWING:**

class carine //class name

{

public static void main(String[] args) //main method of the class

{

int n = 10;

int m = 12;

if(n+m < 10)

{

System.out.println("n + m is less than 10");

}

else

{

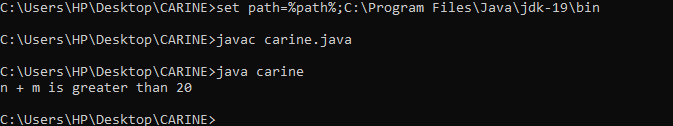
System.out.println("n + m is greater than 20");

}

}

}

**OUTPUT OF THE ABOVE SOURCE CODES IS THE FOLLOWING:**



EXPERIMENT 2: if else if ladder

**Source codes on if else if ladder are the following:**

**class Student //class name**

**{**

**public static void main(String[] args) //main method**

**{**

**String city = "CAIRO";**

**if(city == "Meerut")**

**{**

**System.out.println("city is meerut");**

**}**

**else if (city == "Noida")**

**{**

**System.out.println("city is noida");**

**}**

**else if(city == "Agra")**

**{**

**System.out.println("city is agra");**

**}**

**else**

**{**

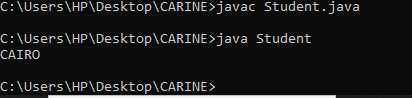
**System.out.println(city);**

**}**

**}**

**}**

**OUTPUT OF THE ABOVE SOURCE CODE IS THE FOLLOWING:**

****

**EXPERIMENT 3: java for each loop**

**SOURCE CODES ARE**

**class BIT //class name**

**{**

**public static void main(String[] args) //main method**

**{**

**String[] names = {"Java","C","C++","Python","JavaScript"};**

**System.out.println("Printing the content of the array names:\n");**

**for(String name:names)**

**{**

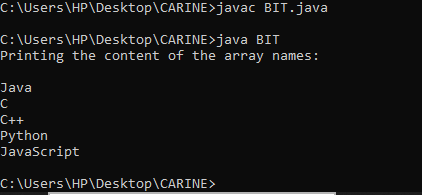
**System.out.println(name);**

**}**

**}**

**}**

**OUTPUT**

****

**SECOND TOPIC:JAVA OBJECT CLASS**

**FIRST EXPERIMENT: object and class example**

**Source codes on java object class within class are the following:**

class object //class name

{

int id;//field or data member or instance variable

String name;

//creating main method inside the Student class

public static void main(String args[])

{

//Creating an object or instance

object s1=new object();//creating an object of Student

//Printing values of the object

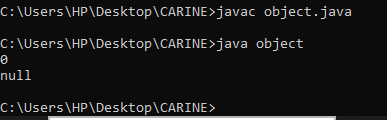
System.out.println(s1.id);//accessing member through reference variable

System.out.println(s1.name);

}

}

Output



**SECOND EXPERIMENT: OBJECT AND CLASS OUTSIDE THE CLASS**

**SOURCE CODES ARE THE FOLLOWING**:

class TEST

{

int id;

String name;

}

//Creating another class TestStudent1 which contains the main method

class TestStudent1

{

public static void main(String args[])

{

TEST OBJECT=new TEST();

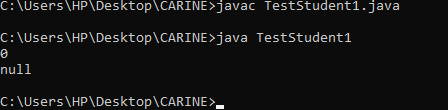
System.out.println(OBJECT.id);

System.out.println(OBJECT.name);

}

}

Output



**Third experiment: object and class with initialization through method**

**Source codes are the following**:

class PRINCE //class name

{

int rollno; //variable name

String name; //variable name

void insertRecord(int r, String n) //main method

{

rollno=r;

name=n;

}

void displayInformation(){System.out.println(rollno+" "+name);

}

}

class TestStudent4

{

public static void main(String args[])

{

PRINCE OB=new PRINCE();

OB.insertRecord(111,"Karan");

OB.insertRecord(222,"Aryan");

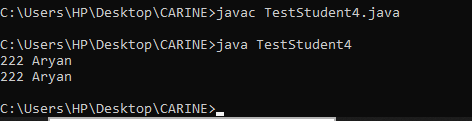
OB.displayInformation();

OB.displayInformation();

}

}

OUTPUT



THIRD TOPIC: INHERITANCE

**SOURCE CODES ON SINGLE INHERITANCE ARE THE FOLLOWING:**

class B

{

void eat()

{

System.out.println("very interesting...");

}

}

class Dog extends B

{

void bark()

{

System.out.println("see you soon...");

}

}

class TestInheritance

{

public static void main(String args[])

{

Dog object=new Dog();

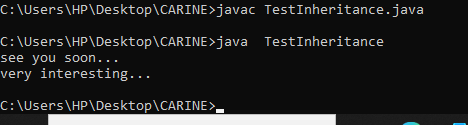
object.bark();

object.eat();

}

}

Output



**SECOND INHERITANCE: MULTILEVEL INHERITANCE**

**Source codes on multilevel inheritance are the following:**

class Animal

{

void eat()

{

System.out.println("thank you...");

}

}

class Dog extends Animal

{

void bark()

{

System.out.println("carine...");

}

}

class BabyDog extends Dog

{

void weep()

{

System.out.println("mutesayire...");

}

}

class TestInheritance2

{

public static void main(String args[])

{

BabyDog d=new BabyDog();

d.weep();

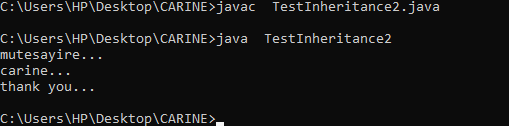
d.bark();

d.eat();

}

}

Output



THIRD EXPERIMENT: hierarchical inheritance

**Source codes on hierarchical inheritance are the following:**

class Animal

{

void eat(){System.out.println("eating...");

}

}

class Dog extends Animal{

void bark(){System.out.println("barking...");

}

}

class Cat extends Animal

{

void meow(){System.out.println("meowing...");

}

}

class TestInheritance3

{

public static void main(String args[])

{

Cat c=new Cat();

c.meow();

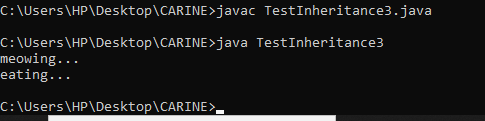
c.eat();

//c.bark();//C.T.Error

}

}

Output



FOURTH TOPIC: JAVA POLYMORPHISM

FIRST EXPERIMENT**: METHOD OVERLOADINF AND CHANGING NO OF ARGUMENTS**

**SOURCE CODES ARE THE FOLLOWING**

class Adder //class name

{

static int add(int a,int b){return a+b;}

static int add(int a,int b,int c){return a+b+c;

}

}

class TestOverloading1

{

public static void main(String[] args) //main method

{

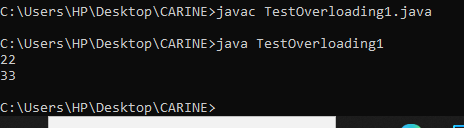
System.out.println(Adder.add(11,11)); //statement

System.out.println(Adder.add(11,11,11));

}

}

Output



SECOND EXPERIMENT: changing data types of arguments

SOURCE CODES ARE THE FOLLOWING:

class Adder //class name

{

static int add(int a, int b){return a+b;}

static double add(double a, double b){return a+b;}

}

class TestOverloading2

{

public static void main(String[] args) //main method

{

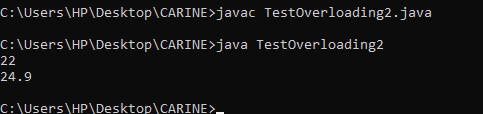
System.out.println(Adder.add(11,11)); //statement 1

System.out.println(Adder.add(12.3,12.6)); //statement 2

}

}

Output



### THIRD EXPERIMENT Example of Method Overloading with Type Promotion if matching found

**SOURCE CODES ARE THE FOLLOWING:**

**class OverloadingCalculation2**

**{**

**void sum(int a,int b)**

**{**

**System.out.println("int arg method invoked");**

**}**

**void sum(long a,long b)**

**{**

**System.out.println("long arg method invoked");**

**}**

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

**{**

**OverloadingCalculation2 obj=new OverloadingCalculation2();**

**obj.sum(20,20);//now int arg sum() method gets invoked**

**}**

**}**

OUTPUT:

