2ND INTERNAL TEST

(M.C.A)

Name: BIBRATA BARMAN

Registration Number: 19352011

Subject: OOPS 2nd internal

ANSWER ANY 2 (15*2)=30

1. Explain any two types of inheritance and write a java program with an example of your choice that involves both these types of inheritance.(in single program)
2. Build a java program to create a text file with entries District names, Corona Infected count, color code. (For example: Chennai, 4970, Red). Through program make few entries of surrounding your districts.
3. Compare and contrast command line interface with GUI by building two versions of the same program with CLI & GUI. Justify your choice of interface with reasons.

Answer:

2. Build a java program to create a text file with entries District names, Corona Infected count, color code. (For example: Chennai, 4970, Red). Through program make few entries of surrounding your districts.

Ans:

Code:

```
/*Write a java program to build a file*/
import java.util.*;
import java.io.*;
class covid19{
	String d_name;
	int c_no;
	String color_code;
	/** Constructs a file object with the given values */
	covid19(String d_name, int c_no,String color_code)
	{
		this.d_name = d_name;
		this.c_no = c_no;
		this.color_code = color_code;
	}
	public static void main(String[] args) throws IOException
	{
```

```
        Scanner s = new Scanner(System.in);
        String awnser="";
        System.out.println("Enter covid_19 Information : ");
        System.out.println("Enter number of district you want to entry data :
");
        int n = s.nextInt();  //Input total number of district
        covid19[] covid_update = new covid19[n];
        FileWriter outfile = null;   //Create files stream out
        try
        {
            outfile = new FileWriter("covid_record.txt"); //Connect the
outfile stream to the "covid_record.txt"
          do
            {
            for(int i=0; i<n; i++)
            {
                s.nextLine();
                System.out.print("District Name : ");
                String d_name = s.nextLine();
                System.out.print("Corona infected count. : ");
                int c_no = s.nextInt();
                System.out.print("District Clour Code(red/green/orange)
: ");
                String color_code = s.next();
```

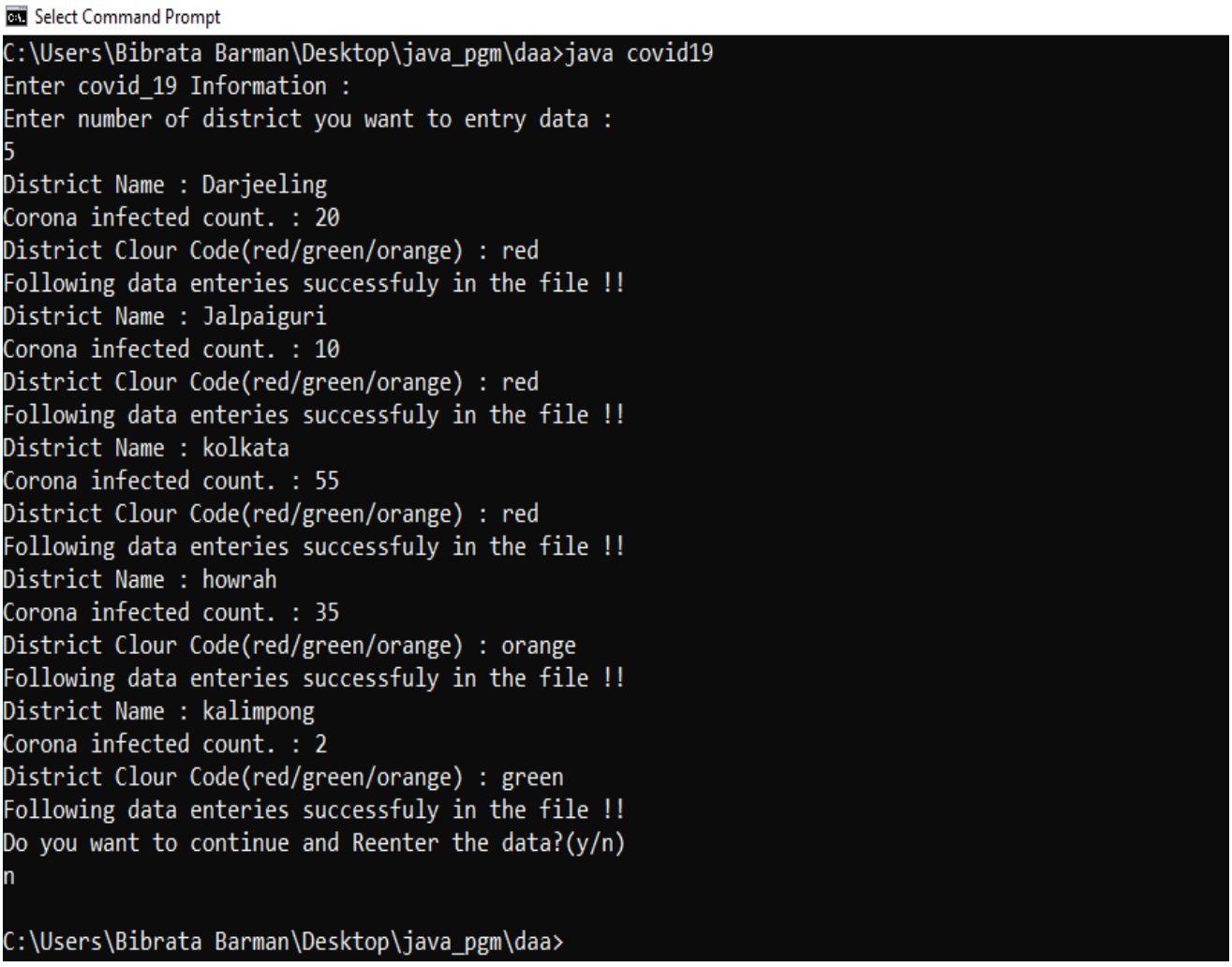
```
                    covid_update[i] = new
covid19(d_name,c_no,color_code);
                    //Write data to the stream
                    outfile.write(covid_update[i].d_name+", ");
                    outfile.write(covid_update[i].c_no+", ");
                    outfile.write(covid_update[i].color_code+"\r\n");
                    System.out.println("Following data enteries successfuly
in the file !!");
                }
                System.out.println("Do you want to continue and Reenter the
data?(y/n)");
                    awnser=s.next();
                }while(awnser.equals("y"));

            }
            catch(IOException e)
            {
                System.out.println("An error occurred in this program !");
                e.printStackTrace();
            }

            finally
            {
                outfile.close();  //close the file
```

```
        }
    } //main
} //class
```

Input/Output:



File (covid_record.txt):

1. Explain any two types of inheritance and write a java program with an example of your choice that involves both these types of inheritance.(in single program)

Ans:

Inheritance:

Inheritance is one of the cornesstone of the object oriented programming because it allows the creation of hierarchical classification. Using inheritance one can create a general class that include some common set of items.

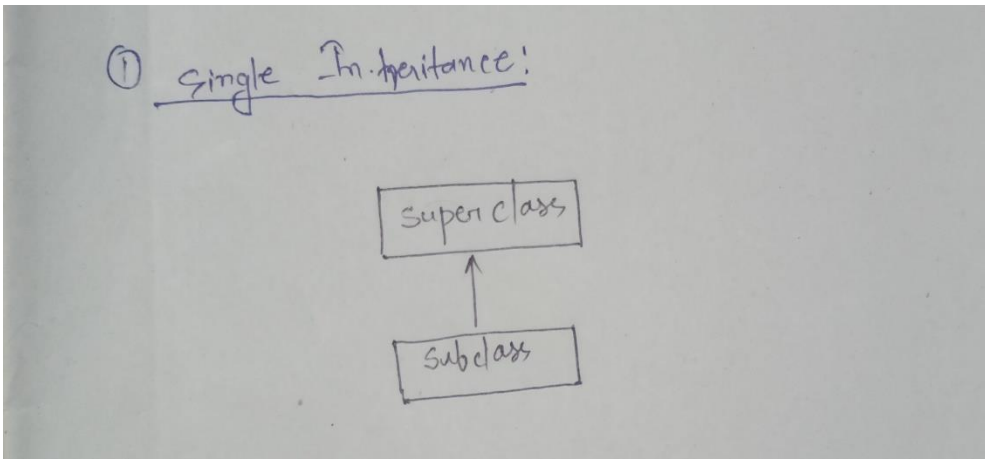
The inheritance allows subclasses to inherit all the variables and methods of their parent classes. There are different types of inheritance –

1. Single inheritance
2. Multiple inheritance (not allowed in java)
3. Hierarchical or multiple single inheritance
4. Multilevel inheritance
5. Hybrid inheritance (not allowed in java)

Java does not directly implement multiple and hybrid inheritance.

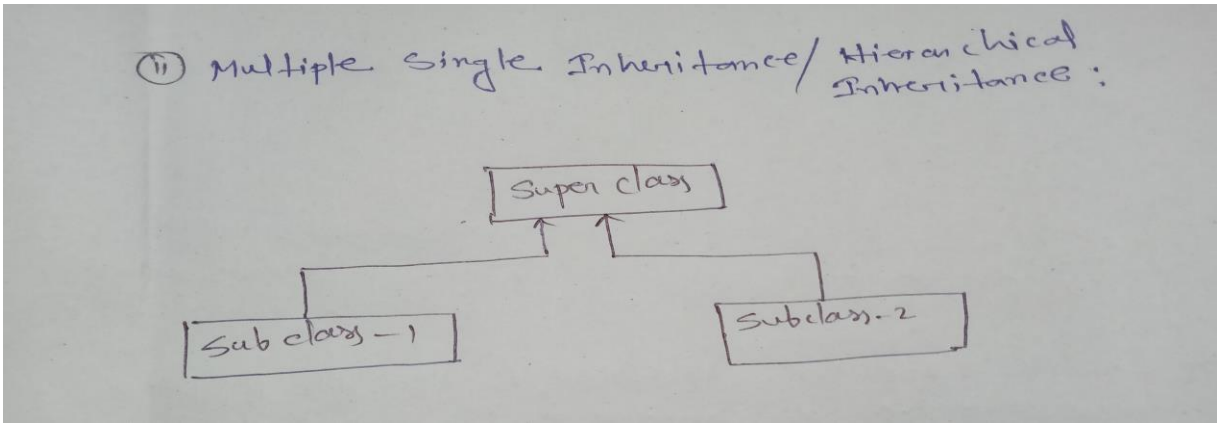
Single inheritance:

When a class extends another one class only then we call it a single inheritance.



Hierarchical or multiple single inheritance :

In such kind of inheritance one class is inherited by many **sub classes**.



Example A java program that involves both these types of inheritance.(in single program):

Source Code:

```
import java.util.Scanner;

class Calculation {     //super class

  int z;

  public void addition(int x, int y) {

    z = x + y;
```

```
      System.out.println("The sum of the given numbers:"+z);
    }
}
class subtraction extends Calculation{      //sub class 1
    public void Subtraction(int x, int y) {
      z = x - y;
      System.out.println("The difference between the given numbers:"+z);
    }
}
class My_Calculation extends Calculation {      //sub class 2
    public void multiplication(int x, int y) {
      z = x * y;
      System.out.println("The product of the given numbers:"+z);
    }
}
class bibro{        //main function
    public static void main(String args[]) {
      String awnser="";
          int choice;
          int a=20 , b=10 ;
          do{
```

```
    Scanner n = new Scanner(System.in);
    Calculation p=new Calculation();
    subtraction p1=new subtraction();
My_Calculation p2= new My_Calculation();
          System.out.println("Menu....");
          System.out.println("1. Single Inheritance ");
          System.out.println("2. Hierarchical Inheritance ");
          System.out.println("Enter the choice : ");
          choice = n.nextInt();
      switch(choice) {
      case 1:   //for class Calculationand & class subtraction
                        p.addition(a,b);
        p1.addition(a,b);
                        p1.Subtraction(a,b);
                        break;
          case 2:   //for class Calculation,subtraction & My_Calculation
                        p.addition(a,b);
                        p1.addition(a,b);
                        p1.Subtraction(a,b);
                        p2.addition(a,b);
                        p2.multiplication(a,b);
```

```
                    break;
            default: System.out.println("Wrong choice..");
            }
            System.out.println("Do you want to continue?(y/n)");
            awnser=n.next();
            }while(awnser.equals("y"));
    }
}
```

Input/Output: