SLIP 1

Q.1 Write a Program to print all Prime numbers in an array of 'n' elements. (use command line arguments)

```
class PrNo
{
 public static void main (String[] args)
 {
 int size = args.length;
 int[] array = new int [size];
 for(int i=0; i<size; i++)
 {
 array[i] = Integer.parseInt(args[i]);
 }
 for(int i=0; i<array.length; i++)
 {
 boolean isPrime = true;
 for (int j=2; j<array[i]; j++)
 {
 if(array[i]%j==0)
 {
 isPrime = false;
 break;
 }
 }
 if(isPrime)
 System.out.println(array[i] + " are the prime numbers in the array ");
 }
 }
}
```

Q.2 Define an abstract class Staff with protected members id and name. Define a parameterized constructor. Define one subclass OfficeStaff with member department. Create n objects of OfficeStaff and display all details.

```
import java.io.*;
import java.util.*;
abstract class staff
{
protected int id;
protected String name;
staff(int id, String name)
```

```
{
this.id=id;
this.name=name;
}
}
class ostaff extends staff
{
String dname;
ostaff(int id,String name, String dname)
{
super(id,name);
this.dname=dname;
}
void display()
{
System.out.println("Staff id is::" +super.id+"\n Staff name
is::"+super.name+"\n Staff department is::"+dname);
}
}
class ademo
{
public static void main(String args[])
{
Scanner sc =new Scanner(System.in);
int n,i,id;
String name,dname;
System.out.println("How many staff you want");
n = sc.nextInt();
ostaff o[]=new ostaff[n];
System.out.println("Enter the staff details");

for(i=0;i<n;i++)
{
System.out.println("Enter the staff id");
id = sc.nextInt();
System.out.println("Enter the staff name");

name = sc.next();

System.out.println("Enter the staff department");

dname = sc.next();

o[i]=new ostaff(id,name,dname);
```

```
}
System.out.println("**********Office staff details**********");
for(i=0;i<n;i++)
o[i].display();

}
}
```

SLIP 2

Q.1 Write a program to read the First Name and Last Name of a person, his weight and height using command line arguments. Calculate the BMI Index which is defined as the individual's body mass divided by the square of their height. (Hint : BMI = Wts. In kgs / (ht)?)

```
import java.util.Scanner;

public class BMICalculator {
   public static void main(String[] args) {
      try (Scanner scanner = new Scanner(System.in)) {
         System.out.print("Enter first name: ");
         String firstName = scanner.nextLine();

         System.out.print("Enter last name: ");
         String lastName = scanner.nextLine();

         System.out.print("Enter weight in kg: ");
         double weight = scanner.nextDouble();

         System.out.print("Enter height in meters: ");
         double height = scanner.nextDouble();

         double bmi = weight / (height * height);

         System.out.println("Name: " + firstName + " " + lastName);
         System.out.println("BMI: " + bmi);
      }
   }
}
```

Q2) Define a class CricketPlayer (name,no_of innings,no_of_times_notout, totatruns, bat_avg). Create an array of n player objects .Calculate the batting average for each

player using static method avg(). Define a static sort method which sorts the array on the basis of average. Display the player details in sorted order.

```
import java.io.*;
import java.util.*;
class cricket
{
String name;
int inn,out,runs,avg;
cricket()
{
name = "Dhoni";
inn = 80;
out = 10;
runs = 5000;
}
cricket(String name,int inn,int out,int runs)
{
this.name= name;
this.inn= inn;
this.out=out;
this.runs=runs;
}
static void avgs(cricket c[])
{
for(int i=0;i<c.length;i++)
c[i].avg = c[i].runs / c[i].inn ;
}
static void sort(cricket c[])
{
int i,j;
cricket c1= new cricket();
for(i=0;i<c.length;i++)
{
for(j=i+1;j<c.length;j++)

{

if(c[i].avg > c[j].avg)
{
c1 = c[i];
c[i]=c[j];
c[j]=c1;
```

```
}
}
}
}
void display()
{
System.out.println("Player name is:: "+name+ "\n Player
innings played are" +inn+ " \n Number of times out " +out+"\n Total runs
are " +runs+"\n Batting average is " +this.avg);
}

public static void main(String args[])
{

Scanner sc =new Scanner(System.in);
System.out.println("********** Default
Information*********");
cricket c1 =new cricket();
c1.display();
int n,i;
String name;
int inn,out,runs,avg;
System.out.println("How many player information you
want");
n = sc.nextInt();
cricket c[] = new cricket[n];
for(i =0;i<n;i++)
{
System.out.println("Enter the player name");

name= sc.next();
System.out.println("Enter the player innings

played");
inn=sc.nextInt();
System.out.println("Enter the player number of
times out");
out=sc.nextInt();

System.out.println("Enter the players total

runs");
runs=sc.nextInt();
c[i]=new cricket(name,inn,out,runs);
```

```
}
System.out.println("**********Player
Information*********");
avgs(c);
for(i =0;i<n;i++)
c[i].display();
sort(c);
System.out.println("**********Player Information after
sorting according to batting avg*********");
for(i =0;i<n;i++)
c[i].display();
}
}
```

SLIP 3

Q1) Write a program to accept ‘n’ name of cities from the user and sort them in ascending order.

```
import java.util.Scanner;

public class CitySorter {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);

      System.out.print("Enter the number of cities: ");
      int n = scanner.nextInt();

      String[] cities = new String[n];

      System.out.println("Enter the city names:");
      for (int i = 0; i < n; i++) {
         cities[i] = scanner.nextLine();
      }
      java.util.Arrays.sort(cities);

      System.out.println("Sorted city names:");
      for (String city : cities) {
         System.out.println(city);
      }
   }
}
```

Q2) Define a class patient (patient_name, patient_age, patient_oxy_level patient HRCT report). Create an object of patient. Handle appropriate exception while patient oxygen level less than 95% and HRCT scan report greater than 10, then throw user defined Exception “Patient 1s Covid Positive(+) and Need to Hospitalized™ otherwise display its information.

```
import java.io.*;

import java.util.*;
class CovidException extends Exception
{
CovidException()
{
System.out.print("Patient is Covid Positive, need to be

hospitalized::");
}
}
class patient
{
String pname;
int age,olevel,hrct;
patient(String pname,int age,int olevel,int hrct)
{
this.pname=pname;
this.age=age;
this.olevel=olevel;
this.hrct=hrct;
}
public static void main(String args[])
{
String pname;
int age,olevel,hrct;
Scanner sc = new Scanner(System.in);
try
{
System.out.println("Enter Patient name");
pname = sc.next();
System.out.println("Enter Patient age");

age = sc.nextInt();

System.out.println("Enter Patient Oxygen level");
```

```
olevel = sc.nextInt();

System.out.println("Enter Patient HRCT scan report");

hrct = sc.nextInt();

patient p =new patient(pname,age,olevel,hrct);
if(olevel < 95 && hrct > 10)
throw new CovidException();
else
System.out.println("Patient name
is::"+pname+"\n Patient age is::"+age+"\n Patient olevel is::"+
olevel+"\n Patient HRCT is::"+ hrct);

}
catch(CovidException e)
{
System.out.println(e);
}
}

}
```

SLIP 4

Q1) Write a program to print an array after changing the rows and columns of a given two-dimensional array.

```
import java.util.*;
class ArrTrans
{
 public static void main(String args[])
 {
 System.out.println("enter the row and column");
 Scanner sc = new Scanner(System.in);
 int r = sc.nextInt();
 int c = sc.nextInt();
 int mat[][] = new int[r][c];
 System.out.println("enter the array elts:");
 for(int i=0;i<r;i++)
 {
 for(int j=0;j<c;j++)
```

```
{

mat[i][j] = sc.nextInt();
}
}
System.out.println("the matrix is:");
for(int i=0;i<c;i++)
{
```

```
for(int j=0;j<r;j++)
{
System.out.print(" " +mat[j][i]);
}
System.out.println(" ");
}
}
```

Q2) Write a program to design a screen using Awt that will take a user name and password. If the user name and password are not same, raise an Exception with appropriate message. User can have 3 login chances only. Use clear button to clear the TextFields.

```
import java.awt.*;
class ldemo extends Frame
{
public static void main(String args[])
{
ldemo l= new ldemo();
l.setLayout(new FlowLayout());
Label l1,l2;
TextField t1,t2;
l1= new Label("Login_ID",Label.RIGHT);
l2= new Label("Password",Label.LEFT);
t1=new TextField(12);
t2=new TextField(8);
t2.setEchoChar('*');
l.add(l1);
l.add(t1);
l.add(l2);
l.add(t2);

l.setTitle("LABLE and TEXTBOX DEMO");
l.setSize(400,150);
```

```
l.setVisible(true);

}

}
```

SLIP 5

Q1) Write a program for multilevel inheritance such that Country is inherited from Continent. State is inherited from Country. Display the place, State, Country and Continent.

```
import java.io.*;
import java.util.*;
class continent
{
String cont;
Scanner sc =new Scanner(System.in);
void cont_input()
{
System.out.println("Enter the continent name");
cont = sc.next();
}
}
class country extends continent
{
String con;
Scanner sc =new Scanner(System.in);
void con_input()
{
System.out.println("Enter the contry name");

con = sc.next();

}
}
class state extends country
{

String sta;
Scanner sc =new Scanner(System.in);
void sta_input()
{
System.out.println("Enter the State name");
```

```
sta = sc.next();
}
}
class place extends state
{
String pla;
Scanner sc =new Scanner(System.in);
void pla_input()
{
System.out.println("Enter the Place name");

pla = sc.next();

}
}
class indemo2 extends place
{
public static void main(String args[])
{
indemo2 p = new indemo2();
p.cont_input();
p.con_input();
p.sta_input();
p.pla_input();

System.out.println("Continent name is :: " + p.cont);
System.out.println("Country name is :: " + p.con);
System.out.println("State name is :: " + p.sta);
System.out.println("Place name is :: " + p.pla);
}
}
```

Q2) Write a menu driven program to perform the following operations on multidimensional array ie matrices : = Addition = Multiplication = Exit

```
import java.util.*;
class Matrix
{
 Scanner sc = new Scanner(System.in);
 int a = sc.nextInt();
 int b = sc.nextInt();
 int M[][] = new int[a][b];
 void accept()
```

```
{
MOBILE: 9730381255 | WWW.NRCLASSESPUNE.COM | WWW.BCSBCA.COM int a =
this.a;
int b = this.b;
System.out.println("enter the "+(a*b)+ " values to matrix:");
for(int i=0;i<a;i++)
{
for(int j=0;j<b;j++)
{
this.M[i][j] = sc.nextInt();
}

}
}
void display()
{
for(int i =0;i<a;i++)
{
for(int j =0;j<b;j++)
{
System.out.print(" "+this.M[i][j]);
}
System.out.println(" ");
}
}
public static void main(String a[])
{
System.out.println("enter size 2*2 or 3*3 or ...");
Matrix m1 = new Matrix();
m1.accept();
System.out.println("values to matrix 1:");
m1.display();
System.out.println("enter the size:");
Matrix m2 = new Matrix();
m2.accept();
System.out.println("values to matrix 2:");
m2.display();

int choice;
Scanner scanner = new Scanner(System.in);
while(true) {
System.out.println("Press 1: Addition, 2: Multiplication, 3: Exit");
choice = scanner.nextInt();
switch (choice) {
```

```
 case 1:
 System.out.println("Addition is:" );
 for(int i=0;i<m1.a;i++)
 {
 for(int j=0;j<m1.b;j++)
 {
 System.out.print(" "+ (m1.M[i][j]+m2.M[i][j]));
 MOBILE: 9730381255 | WWW.NRCLASSESPUNE.COM | WWW.BCSBCA.COM }
 System.out.println(" ");
 }
 break;
 case 2:
 System.out.println("Multiplication is:");
 for(int i=0;i<m2.a;i++)
 {
 for(int j=0;j<m2.b;j++)
 {
 System.out.print(" "+
(m1.M[i][j]*m2.M[i][j]));
 }
 System.out.println(" ");
 }
 break;

 case 3:
 System.exit(0);
 }
 }
 }
}
```

SLIP 6

Q1) Write a program to display the Employee(Empid, Empname, Empdesignation, Empsal) information using toString().

```
import java.io.*;
import java.util.*;
class emp
{
int eid;
String name,dname;
float salary;
```

```
static int cnt =0;

emp()
{
cnt ++;
}
emp(int eid,String name, String dname, float salary)
{
this.eid= eid;
this.name=name;
this.dname = dname;
this.salary = salary;
cnt++;
}
void display()
{
System.out.println("emp id is " +eid +"\n Employee name is :" +
name + "\n Employee department is " +dname +"\n Employee salary is " +salary);
}
static void count()
{
System.out.println("Number of objects created are " +cnt);
}

public static void main(String args[])
{
int i,n,id;
String name,dname;
float salary;
Scanner sc =new Scanner(System.in);
System.out.println("****** Default constructor values
are**********");
emp e1 = new emp();
e1.display();
System.out.println("How many employee you want");
n = sc.nextInt();
emp e[] = new emp[n];
for(i=0;i<n;i++)
{
System.out.println("Enter the id");
id = sc.nextInt();
System.out.println("Enter the name");
name = sc.next();
System.out.println("Enter the department name");
```

```
dname = sc.next();
System.out.println("Enter the salary");
salary = sc.nextFloat();

e[i] = new emp(id,name,dname,salary);
}
for(i=0;i<n;i++)
e[i].display();
e1.count();
}
}
```

Q2) Create an abstract class “order” having members id, description. Create two subclasses “PurchaseOrder” and “Sales Order” having members customer name and Vendor name respectively. Definemethods accept and display in all cases. Create 3 objects each of Purchase Order and Sales Order and accept and display details.

```
import java.io.*;
import java.util.*;
abstract class order
{
int id;
String des;
}
class porder extends order
{
String cname;
porder(int id, String des, String cname)
{
super.id=id;
super.des=des;
this.cname=cname;
}
void display()
{
System.out.println("ID is::"+super.id+"\n Description

is::"+super.des+"\n Customer name is::"+cname);
}

}
class sorder extends order
{
String vname;
```

```
sorder(int id, String des, String vname)
{
super.id=id;
super.des=des;
this.vname=vname;
}
void display()
{
System.out.println("ID is::"+super.id+"\n Description
is::"+super.des+"\n Vendor name is::"+vname);
}
}
public class pdemo
{
public static void main(String args[])
{
int i,id;
String des,cname,vname;
sorder s[]=new sorder[3];
porder p[]=new porder[3];
Scanner sc =new Scanner(System.in);
System.out.println("*****Enter purchase order details******");
for(i=0;i<3;i++)
{
System.out.println("Enter id");
id=sc.nextInt();
System.out.println("Enter description");

des=sc.next();

System.out.println("Enter cusomer name");

cname=sc.next();

p[i]=new porder(id,des,cname);
}
System.out.println("*****Enter Sales order details******");

for(i=0;i<3;i++)
{
System.out.println("Enter id");
id=sc.nextInt();
System.out.println("Enter description");
des=sc.next();
```

```
System.out.println("Enter vendor name");
vname=sc.next();
s[i]=new sorder(id,des,vname);
}

System.out.println("*****Purchase order details******");

for(i=0;i<3;i++)
p[i].display();
System.out.println("*****Sales order details******");

for(i=0;i<3;i++)
s[i].display();
}
}
```

SLIP 7

Q1) Design a class for Bank. Bank Class should support following operations; a. Deposit a certain amount into an account b. Withdraw a certain amount from an account c. Return a Balance value specifying the amount with details

```
import java.io.*;
import java.util.*;
class bank
{
double balance;

bank()
{
balance = 0;
}
bank(double inbalance)
{
balance = inbalance;
}
public void deposite(double amount)
{
balance = balance + amount;
}
public void withdraw(double amount)
{
balance = balance - amount;
}
```

```
public double getbalance()
{
return balance;
}
public static void main(String args[])
{
bank b=new bank(1000);
b.withdraw(250);
System.out.println("After the withdraw balance is::"
+b.balance);
b.deposite(650);
System.out.println("After the deposite balance is::"
+b.balance);
System.out.println("After the all the transction balance
is::" +b.getbalance());
}
}
```

Q2) Write a program to accept a text file from user and display the contents of a file in reverse order and change its case.

```
import java.io.*;
import java.util.*;

public class FileReverser {
   public static void main(String[] args) throws IOException {
      Scanner scanner = new Scanner(System.in);

      System.out.print("Enter the file name: ");
      String fileName = scanner.nextLine();

            List<String> lines = new ArrayList<>();
      try (BufferedReader reader = new BufferedReader(new FileReader(fileName))) {
         String line;
         while ((line = reader.readLine()) != null) {
            lines.add(line);
         }
      }

         Collections.reverse(lines);
      for (int i = 0; i < lines.size(); i++) {
         lines.set(i, reverseCase(lines.get(i)));
      }
```

```
        try (BufferedWriter writer = new BufferedWriter(new FileWriter(fileName))) {
            for (String line : lines) {
                writer.write(line);
                writer.newLine();
            }
        }

        System.out.println("File contents reversed and case changed.");
    }

    private static String reverseCase(String str) {
        StringBuilder sb = new StringBuilder();
        for (char c : str.toCharArray()) {
            if (Character.isUpperCase(c)) {
                sb.append(Character.toLowerCase(c));
            } else {
                sb.append(Character.toUpperCase(c));
            }
        }
        return sb.toString();
    }
}
```

SLIP 8

Q1) Create a class Sphere, to calculate the volume and surface area of sphere. (Hint : Surface area=4*3.14(r*r), Volume=(4/3)3.14(r*r*r))

```
class Sphere {
    private double radius;

    public Sphere(double radius) {
        this.radius = radius;
    }

    public double calculateSurfaceArea() {
        return 4 * Math.PI * Math.pow(radius, 2);
    }

    public double calculateVolume() {
        return (4.0 / 3.0) * Math.PI * Math.pow(radius, 3);
    }
```

```
}

public class SphereCalculator {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the radius of the sphere: ");
        double radius = scanner.nextDouble();

        Sphere sphere = new Sphere(radius);

        double surfaceArea = sphere.calculateSurfaceArea();
        double volume = sphere.calculateVolume();

        System.out.println("Surface area: " + surfaceArea);
        System.out.println("Volume: " + volume);
    }
}
```

Q2) Design a screen to handle the Mouse Events such as MOUSE_MOVED and MOUSE_CLICKED and display the position of the Mouse_Click in a TextField.

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class MouseEventDemo extends JFrame {
    private JLabel label;
    private JTextField textField;

    public MouseEventDemo() {
        super("Mouse Event Demo");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new BorderLayout());

        label = new JLabel("Move the mouse or click inside this label");
        add(label, BorderLayout.CENTER);

        textField = new JTextField(30);
        add(textField, BorderLayout.SOUTH);

        label.addMouseListener(new MouseAdapter() {
            public void mouseClicked(MouseEvent e) {
```

```
            textField.setText("Mouse clicked at: (" + e.getX() + ", " + e.getY() + ")");
        }

        public void mouseMoved(MouseEvent e) {
            label.setToolTipText("Mouse moved to: (" + e.getX() + ", " + e.getY() + ")");
        }
    });

    setSize(400, 200);
    setVisible(true);
  }

  public static void main(String[] args) {
    new MouseEventDemo();
  }
}
```

SLIP 9

Q.1 Define a "Clock™ class that does the following ; a. Accept Hours, Minutes and Seconds b. Check the validity of numbers c. Set the time to AM/PM mode Use the necessary constructors and methods to do the above task

```
import java.io.*;
import java.util.*;
class clock
{
int h,m,s;
clock()
{
Scanner sc = new Scanner(System.in);
System.out.println("Enter the hours");
h = sc.nextInt();
System.out.println("Enter the mins");
m = sc.nextInt();
System.out.println("Enter the secs");
s = sc.nextInt();

}
void isTimevalid()
{
if(h>=0 && h<=24 && m>=0 && m<=60 && s>=0 && m<=60)
System.out.println("Time is Valid");
else
```

```
System.out.println("Time is not Valid");
}
void setTime()
{
if(h < 12)
System.out.println(" Time is:: "+h+":"+m+":"+s+
"AM");
else
{
h=h-12;

System.out.println(" Time is:: "+h+":"+m+":"+s+

"PM");
}
}
public static void main(String args[])
{
clock c = new clock();
c.isTimevalid();
c.setTime();
}
}
```

Q2) Write a program to using marker interface create a class Product (product_id, product_name, product_cost, product_quantity) default and parameterized constructor. Create objectsof class product and display the contents of each object and Also display the object count. [

```
import java.util.*;
interface markerint
{
}
public class product implements markerint
{
int pid;
String pname;
double cost;
static int cnt =0;
product()
{

pid =101;
pname="Pen";
```

```
cost=20;
cnt++;

}
product(int pid, String pname,double cost)
{
this.pid =pid;
this.pname=pname;
this.cost=cost;
cnt++;
}
void display()
{
System.out.println("Product id is::" +pid+ "\n Product name is::"

+pname + "\n Product Cost is::" +cost + "\n Object Count is::" +cnt);
}
public static void main(String args[]) throws
{
int pid,n,i;
String pname;
double cost;

product p = new product();

System.out.println("********* Default Counstructor
Information*********");
p.display();

Scanner sc = new Scanner(System.in);
System.out.println("How many product you want");
n = sc.nextInt();
product p1[] = new product[n];
for(i=0;i<n;i++)
{
System.out.println("Enter Product id");
pid = sc.nextInt();

System.out.println("Enter Product name");
pname = sc.next();

System.out.println("Enter Product Cost");

cost = sc.nextDouble();
```

```
p1[i] = new product(pid,pname,cost);
}
System.out.println("********* Parameterised Counstructor

Information*********");
for(i=0;i<n;i++)
p1[i].display();
}

}
```

SLIP 10

Q.1) Write a program to find the cube of given number using functional interface.

```
import java.io.*;
import java.util.*;
interface cube
{
int cubecal();
}
class idemo implements cube
{
public int cubecal()
{
Scanner sc =new Scanner(System.in);
int n;
System.out.println("Enter the number");
n = sc.nextInt();
return n*n*n;
}
public static void main(String args[])
{
idemo i=new idemo();
System.out.println("Cube of a number is::" +i.cubecal());
}
}
```

Q2) Write a program to create a package name student. Define class StudentInfo with method to display information about student such as rollno, class, and percentage. Create another class StudentPer with method to find percentage of the student. Accept student details like rollno, name, class and marks of 6 subject from user.

```
import mypack.*;
import java.io.*;
public class pdemo
{
public static void main(String args[])

{
demo d=new demo();
d.display();
}

}
```

Step 5: compile the file (javac pdemo.java)
Step 6: Run the file (java pdemo)

Qu 5 Student result display package program

```
package SY;
import java.util.*;
public class syclass
{
public int ct,mt,et;
public void getdata()
{
Scanner sc = new Scanner(System.in);
System.out.println("Enter the marks of computer out off

100");

ct = sc.nextInt();
System.out.println("Enter the marks of maths out off

100");
mt = sc.nextInt();

System.out.println("Enter the marks of electronics out

off 100");
et = sc.nextInt();

}
```

```
}

package TY;
import java.util.*;
public class tyclass
{
public int th,prac;
public void getdata()
{
Scanner sc = new Scanner(System.in);
System.out.println("Enter the marks of theory out off

500");

th = sc.nextInt();
System.out.println("Enter the marks of practical out off

500");
prac = sc.nextInt();

}

}

import SY.*;
import TY.*;
import java.util.*;
class student
{
int rno,syt,tyt,gt;
String name,grade;
float per;
public void getdata()
{
Scanner sc = new Scanner(System.in);

System.out.println("Enter the roll number");
rno = sc.nextInt();
System.out.println("Enter the name of student");
name = sc.next();

}
}
class studentinfo
```

```
{
public static void main(String args[])
{
Scanner sc = new Scanner(System.in);
int i,n;
System.out.println("How many student you want");
n=sc .nextInt();

student si[] = new student[n];
syclass s[] =new syclass[n];
tyclass t[]= new tyclass[n];
System.out.println("Enter " +n+ "records");
for(i=0;i<n;i++)
{
si[i]=new student();
s[i]=new syclass();
t[i]= new tyclass();
si[i].getdata();
s[i].getdata();
t[i].getdata();

si[i].syt=s[i].ct + s[i].mt + s[i].et;
si[i].tyt = t[i].th + t[i].prac;
si[i].gt = si[i].syt + si[i].tyt;
si[i].per = si[i].gt / 13;
if(si[i].per >= 70)
si[i].grade="A";
else if(si[i].per < 70 && si[i].per >=60 )

si[i].grade="B";
else if(si[i].per < 60 && si[i].per >=50)
si[i].grade="C";
else if(si[i].per < 50 && si[i].per >=40)
si[i].grade="PASS";
else
si[i].grade = "Fail";
}
System.out.println("**********Student
Information**********");
System.out.println("Roll number \t Name \t SY total \t TY
total \t Grand total \t Percenatge \t Grade");
for(i=0;i<n; i++)
System.out.println(si[i].rno + "\t" + si[i].name +"\t" +
si[i].syt + "\t" + si[i].tyt +"\t" +si[i].gt +"\t" +si[i].per + "\t" +
```

```
si[i].grade);
}
}
```

Slip 11

Q.1)Define an interface “Operation” which has method volume( ).Define a constant PI having a value 3.142 Create a class cylinder which implements this interface (members — radius,height). Create one object and calculate the volume.

```
interface Operation {
   double PI = 3.142;
   double volume();
}

class Cylinder implements Operation {
   private double radius;
   private double height;

   public Cylinder(double radius, double height) {
      this.radius = radius;
      this.height = height;
   }

   @Override
   public double volume() {
      return PI * radius * radius * height;
   }
}

public class CylinderVolume {
   public static void main(String[] args) {
      Cylinder cylinder = new Cylinder(5, 10);
      double volume = cylinder.volume();
      System.out.println("Volume of the cylinder: " + volume);
   }
}
```

Q2) Write a program to accept the username and password from user if username and password are not same then raise "Invalid Password" with appropriate msg.

```
import java.io.*;
import java.util.*;
class InvalidUsernameException extends Exception
```

```
{
InvalidUsernameException(String u)
{
System.out.print("\n Invalid Username Exception caught::"

+u);
}
}
class InvalidPasswordException extends Exception
{
InvalidPasswordException(String p)

{

System.out.print("\n Invalid Password Exception caught::"

+p);
}
}
class logindemo
{
String username,password;
logindemo()
{
username="Computer";
password="123";
}
logindemo(String u, String p)
{
this.username=u;
this.password=p;
}
public static void main(String args[])
{
logindemo d= new logindemo();
String u,p;
try
{
u = args[0];
p = args[1];
logindemo d1= new logindemo(u,p);
if(d.username.equals(d1.username))
System.out.println("Username is Valid");
else
```

```
throw new InvalidUsernameException(u);

if(d.password.equals(d1.password))

System.out.println("Password is Valid");
else
throw new InvalidPasswordException(p);

}
catch(InvalidUsernameException uu)
{
System.out.print("\t" + uu);
}
catch(InvalidPasswordException pp)

{

System.out.print("\t" +pp);

}
}
}
```

SLIP 12

Q1) Write a program to create parent class College(cno, cname, caddr) and derived class Department(dno, dname) from College. Write a necessary methods to display College details.

```
import java.io.*;
import java.util.*;
class college
{
int cno;
String cname,cadd;
}
class dept extends college
{
int dno;
String dname;

public void accept()
{
Scanner sc = new Scanner(System.in);
```

```
System.out.println("Enter the college code");

super.cno=sc.nextInt();

System.out.println("Enter the college name");

super.cname=sc.next();

System.out.println("Enter the college address");

super.cadd=sc.next();
System.out.println("Enter the Department code");

dno=sc.nextInt();

System.out.println("Enter the department name");
dname=sc.next();
}
public void display()
{
System.out.println("College code ::" +super.cno);
System.out.println("College name::" + super.cname);
System.out.println("College address::" + super.cadd);

System.out.println("Department code::" +dno);
System.out.println("Department name::" +dname);

}
public static void main(String args[])
{
dept d =new dept();
d.accept();
d.display();
}
}
```

Q2) Write a java program that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the +, -, *, % operations. Add a text field to display the result.

```
 import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
```

```
import java.awt.event.ActionListener;

public class SimpleCalculator extends JFrame implements ActionListener {
    private JTextField display;
    private String operator;
    private double firstOperand, secondOperand;

    public SimpleCalculator() {
        setTitle("Simple Calculator");
        setSize(400, 400);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setLayout(new BorderLayout());

        // Create display field
        display = new JTextField();
        display.setEditable(false);
        display.setHorizontalAlignment(JTextField.RIGHT);
        add(display, BorderLayout.NORTH);

        // Create panel for buttons
        JPanel buttonPanel = new JPanel();
        buttonPanel.setLayout(new GridLayout(4, 4, 5, 5));

        // Button array
        String[] buttons = {
            "7", "8", "9", "/",
            "4", "5", "6", "*",
            "1", "2", "3", "-",
            "0", "C", "=", "+"
        };

        // Add buttons to panel
        for (String text : buttons) {
            JButton button = new JButton(text);
            button.addActionListener(this);
            buttonPanel.add(button);
        }

        add(buttonPanel, BorderLayout.CENTER);
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        String command = e.getActionCommand();
```

```
        if (command.charAt(0) >= '0' && command.charAt(0) <= '9') {
            // If a digit is pressed, append it to the display
            display.setText(display.getText() + command);
        } else if (command.equals("C")) {
            // Clear the display
            display.setText("");
        } else if (command.equals("=")) {
            // Calculate and display the result
            secondOperand = Double.parseDouble(display.getText());
            double result = calculate(firstOperand, secondOperand, operator);
            display.setText(String.valueOf(result));
        } else {
            // Store the first operand and operator
            if (!display.getText().isEmpty()) {
                firstOperand = Double.parseDouble(display.getText());
            }
            operator = command;
            display.setText(""); // Clear the display for next number
        }
    }

    private double calculate(double first, double second, String op) {
        switch (op) {
            case "+":
                return first + second;
            case "-":
                return first - second;
            case "*":
                return first * second;
            case "/":
                return second != 0 ? first / second : 0; // Avoid division by zero
            default:
                return 0;
        }
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> {
            SimpleCalculator calculator = new SimpleCalculator();
            calculator.setVisible(true);
        });
    }
}
```

SLIP 13

Q1) Write a program to accept a file name from command prompt, if the file exits then display number of words and lines in that file.

```
import java.io.*;

public class WordAndLineCount {
   public static void main(String[] args) throws IOException {
      if (args.length != 1) {
         System.out.println("Usage: java WordAndLineCount <filename>");
         return;
      }

      try (BufferedReader reader = new BufferedReader(new FileReader(args[0]))) {
         int lines = 0, words = 0;
         for (String line; (line = reader.readLine()) != null;) {
            lines++;
            words += line.split("\\s+").length;
         }
         System.out.println("Lines: " + lines + ", Words: " + words);
      } catch (IOException e) {
         System.out.println("Error: " + e.getMessage());
      }
   }
}
```

Q2) Write a program to display the system date and time in various formats shown below: Current date 1s : 31/08/2021 Current date is : 08-31-2021 Current date is : Tuesday August 31 2021 Current date and time is : Fri August 31 15:25:59 IST 2021 Current date and time is : 31/08/21 15:25:59 PM +0530

```
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.TimeZone;

public class DateFormatExample {
   public static void main(String[] args) {
      // Get the current date and time
      Date now = new Date();

      // Define the date formats
```

```
        SimpleDateFormat format1 = new SimpleDateFormat("dd/MM/yyyy");
        SimpleDateFormat format2 = new SimpleDateFormat("MM-dd-yyyy");
        SimpleDateFormat format3 = new SimpleDateFormat("EEEE MMMM dd yyyy");
        SimpleDateFormat format4 = new SimpleDateFormat("EEE MMMM dd HH:mm:ss z
yyyy");
        SimpleDateFormat format5 = new SimpleDateFormat("dd/MM/yy hh:mm:ss a Z");

        // Set the timezone for format4 and format5 to IST
        format4.setTimeZone(TimeZone.getTimeZone("IST"));
        format5.setTimeZone(TimeZone.getTimeZone("IST"));

        // Display the dates in the specified formats
        System.out.println("Current date is : " + format1.format(now));
        System.out.println("Current date is : " + format2.format(now));
        System.out.println("Current date is : " + format3.format(now));
        System.out.println("Current date and time is : " + format4.format(now));
        System.out.println("Current date and time is : " + format5.format(now));
    }
}
```

SLIP 14

Q1) Write a program to accept a number from the user, if number is zero then throw user defined exception “Number is 0” otherwise check whether no is prime or not (Use static keyword).

```
import java.util.Scanner;

// User-defined exception
class ZeroException extends Exception {
    public ZeroException(String message) {
        super(message);
    }
}

public class PrimeChecker {

    // Static method to check if a number is prime
    static boolean isPrime(int number) {
        if (number <= 1) {
            return false; // 0 and 1 are not prime numbers
        }
        for (int i = 2; i <= Math.sqrt(number); i++) {
            if (number % i == 0) {
```

```
            return false; // Found a divisor, not prime
        }
    }
    return true; // No divisors found, it's prime
}

public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    System.out.print("Enter a number: ");

    try {
        int number = scanner.nextInt();

        // Throw exception if number is zero
        if (number == 0) {
            throw new ZeroException("Number is 0");
        }

        // Check if the number is prime
        if (isPrime(number)) {
            System.out.println(number + " is a prime number.");
        } else {
            System.out.println(number + " is not a prime number.");
        }
    } catch (ZeroException e) {
        System.out.println(e.getMessage());
    } catch (Exception e) {
        System.out.println("Invalid input. Please enter an integer.");
    } finally {
        scanner.close();
    }
}
}
```

Q2) Write a Java program to create a Package “SY” which has a class SYMarks (members — ComputerTotal, MathsTotal, and ElectronicsTotal). Create another package TY which has a class TYMarks (members — Theory, Practicals). Create ‘n” objects of Student class (having rollNumber, name, SYMarks and TYMarks). Add the marks of SY and TY computer subjects and calculate the Grade (A’ for >= 70, ‘B’ for >= 60 ‘C’ for >= 50, Pass Class for > =40 else’'FAIL’) and display the result of the student in proper format.

```
package SY;
import java.util.*;
```

```
public class syclass
{
public int ct,mt,et;
public void getdata()
{
Scanner sc = new Scanner(System.in);
System.out.println("Enter the marks of computer out off

100");

ct = sc.nextInt();
System.out.println("Enter the marks of maths out off

100");
mt = sc.nextInt();

System.out.println("Enter the marks of electronics out

off 100");
et = sc.nextInt();

}

}

package TY;
import java.util.*;
public class tyclass
{
public int th,prac;
public void getdata()
{
Scanner sc = new Scanner(System.in);
System.out.println("Enter the marks of theory out off

500");

th = sc.nextInt();
System.out.println("Enter the marks of practical out off

500");
prac = sc.nextInt();

}
```

```
}

import SY.*;
import TY.*;
import java.util.*;
class student
{
int rno,syt,tyt,gt;
String name,grade;
float per;
public void getdata()
{
Scanner sc = new Scanner(System.in);

System.out.println("Enter the roll number");
rno = sc.nextInt();
System.out.println("Enter the name of student");
name = sc.next();

}
}
class studentinfo
{
public static void main(String args[])
{
Scanner sc = new Scanner(System.in);
int i,n;
System.out.println("How many student you want");
n=sc .nextInt();

student si[] = new student[n];
syclass s[] =new syclass[n];
tyclass t[]= new tyclass[n];
System.out.println("Enter " +n+ "records");
for(i=0;i<n;i++)
{
si[i]=new student();
s[i]=new syclass();
t[i]= new tyclass();
si[i].getdata();
s[i].getdata();
t[i].getdata();
```

```
si[i].syt=s[i].ct + s[i].mt + s[i].et;
si[i].tyt = t[i].th + t[i].prac;
si[i].gt = si[i].syt + si[i].tyt;
si[i].per = si[i].gt / 13;
if(si[i].per >= 70)
si[i].grade="A";
else if(si[i].per < 70 && si[i].per >=60 )

si[i].grade="B";
else if(si[i].per < 60 && si[i].per >=50)
si[i].grade="C";
else if(si[i].per < 50 && si[i].per >=40)
si[i].grade="PASS";
else
si[i].grade = "Fail";
}
System.out.println("***************Student
Information************");
System.out.println("Roll number \t Name \t SY total \t TY
total \t Grand total \t Percenatge \t Grade");
for(i=0;i<n; i++)
System.out.println(si[i].rno + "\t" + si[i].name +"\t" +
si[i].syt + "\t" + si[i].tyt +"\t" +si[i].gt +"\t" +si[i].per + "\t" +
si[i].grade);
}
}
```

SLIP 15

1.Accept the names of two files and copy the contents of the first to the second. First file having Book name and Author name in file.

```
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.Scanner;

public class FileCopy {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
```

```
        System.out.print("Enter the source file name (with path if necessary): ");
        String sourceFileName = scanner.nextLine();

        System.out.print("Enter the destination file name (with path if necessary): ");
        String destinationFileName = scanner.nextLine();

        // Copying contents
        try (BufferedReader reader = new BufferedReader(new FileReader(sourceFileName));
             BufferedWriter writer = new BufferedWriter(new FileWriter(destinationFileName)))
{

            String line;
            while ((line = reader.readLine()) != null) {
                writer.write(line);
                writer.newLine(); // Add a new line after each line copied
            }

            System.out.println("Contents copied successfully from " + sourceFileName + " to
" + destinationFileName);
        } catch (IOException e) {
            System.out.println("An error occurred: " + e.getMessage());
        } finally {
            scanner.close();
        }
    }
}
```

Q2) Write a program to define a class Account having members custname, accno. Define default and parameterized constructor. Create a subclass called SavingAccount with member savingbal, minbal. Create a derived class AccountDetail that extends the class SavingAccount with members, depositamt and withdrawalamt. Write a appropriate method to display customer details.

```
class Account {
    String custname;
    String accno;

    // Default constructor
    public Account() {
        custname = "Unknown";
        accno = "0000";
    }
```

```
    // Parameterized constructor
    public Account(String custname, String accno) {
        this.custname = custname;
        this.accno = accno;
    }
}

class SavingAccount extends Account {
    double savingbal;
    double minbal;

    // Default constructor
    public SavingAccount() {
        super();
        savingbal = 0.0;
        minbal = 500.0; // Example minimum balance
    }

    // Parameterized constructor
    public SavingAccount(String custname, String accno, double savingbal, double
minbal) {
        super(custname, accno);
        this.savingbal = savingbal;
        this.minbal = minbal;
    }
}

class AccountDetail extends SavingAccount {
    double depositamt;
    double withdrawalamt;

    // Default constructor
    public AccountDetail() {
        super();
        depositamt = 0.0;
        withdrawalamt = 0.0;
    }

    // Parameterized constructor
    public AccountDetail(String custname, String accno, double savingbal, double minbal,
double depositamt, double withdrawalamt) {
        super(custname, accno, savingbal, minbal);
        this.depositamt = depositamt;
        this.withdrawalamt = withdrawalamt;
```

```
    }

    // Method to display customer details
    public void displayDetails() {
        System.out.println("Customer Name: " + custname);
        System.out.println("Account Number: " + accno);
        System.out.println("Saving Balance: " + savingbal);
        System.out.println("Minimum Balance: " + minbal);
        System.out.println("Deposit Amount: " + depositamt);
        System.out.println("Withdrawal Amount: " + withdrawalamt);
    }
}

public class Main {
    public static void main(String[] args) {
        AccountDetail accountDetail = new AccountDetail("John Doe", "12345", 1500.0,
500.0, 200.0, 100.0);
        accountDetail.displayDetails();
    }
}
```

SLIP 16

Q.1)Write a program to find the Square of given number using function interface.

```
import java.io.*;
import java.util.*;
interface square
{
int squarecal();
}
class idemo2 implements square
{
public int squarecal()
{
Scanner sc =new Scanner(System.in);
int n;
System.out.println("Enter the number");
n = sc.nextInt();
return n*n;
}
public static void main(String args[])
{
```

```
idemo2 i=new idemo2();
System.out.println("Square of a number is::" +i.squarecal());
}
}
```

Q.2)

SLIP 17

Q.1 Design a Super class Customer (name, phone-number). Derive a class Depositor(accno , balance) from Customer. Again, derive a class Borrower (loan-no, loan-amt) from Depositor. Write necessary member functions to read and display the details of "n'customers

```
 import java.io.*;
import java.util.*;
class customer
{
String name;
int ph;
}
class depositor extends customer
{
int acno;
double balance;
}
class borrower extends depositor
{
int lno;
double lamount;
public void read()
{
Scanner sc =new Scanner(System.in);
System.out.println("Enter the customer name");
super.name = sc.next();
System.out.println("Enter the customer phone number");

super.ph = sc.nextInt();

System.out.println("Enter the depositer account number");

super.acno = sc.nextInt();
```

```
System.out.println("Enter the depositer account balance");

super.balance = sc.nextDouble();

System.out.println("Enter the borrower loan number");

lno = sc.nextInt();

System.out.println("Enter the borrower loan amount");

lamount = sc.nextDouble();
}
public void display()
{
System.out.println("Customer name is::" +super.name + "\n
Phone number ::"+super.ph +"\n Account number is::" +super.acno +"\n
Account balance is::"+super.balance + "\n Loan number is::"+lno+"\n Loan
amount is::"+lamount);
}
}
class cdemo
{
public static void main(String args[])
{
int n,i;
Scanner sc = new Scanner(System.in);
System.out.println("How many customer you want");
n= sc.nextInt();
borrower b[] = new borrower[n];
System.out.println("Enter "+n+" Customer information");
for(i=0;i<n;i++)
{
b[i]=new borrower();
b[i].read();
}
System.out.println("***********Customer

information***********");
for(i=0;i<n;i++)
b[i].display();
}
}
```

Q2) Write Java program to design three text boxes and two buttons using swing. Enter different strings in first and second textbox. On clicking the First command button, concatenation of two strings should be displayed in third text box and on clicking second command button, reverse of string should display in third text box

```
import javax.swing.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class StringManipulator {

    public static void main(String[] args) {
        JFrame frame = new JFrame("String Manipulator");
        frame.setSize(400, 200);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setLayout(null);

        // Text boxes
        JTextField textBox1 = new JTextField();
        JTextField textBox2 = new JTextField();
        JTextField resultBox = new JTextField();
        resultBox.setEditable(false);

        // Set bounds
        textBox1.setBounds(50, 30, 280, 30);
        textBox2.setBounds(50, 70, 280, 30);
        resultBox.setBounds(50, 110, 280, 30);

        // Buttons
        JButton concatButton = new JButton("Concatenate");
        JButton reverseButton = new JButton("Reverse");

        concatButton.setBounds(50, 150, 130, 30);
        reverseButton.setBounds(200, 150, 130, 30);

        // Action listeners
        concatButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                String result = textBox1.getText() + textBox2.getText();
                resultBox.setText(result);
            }
        });
```

```
        reverseButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                String input = textBox1.getText();
                String reversed = new StringBuilder(input).reverse().toString();
                resultBox.setText(reversed);
            }
        });

        // Add components to frame
        frame.add(textBox1);
        frame.add(textBox2);
        frame.add(resultBox);
        frame.add(concatButton);
        frame.add(reverseButton);

        frame.setVisible(true);
    }
}
```

SLIP 18

Q1) Write a program to implement Border Layout Manager.

```
import javax.swing.*;
import java.awt.*;

public class BorderLayoutExample {

    public static void main(String[] args) {
        JFrame frame = new JFrame("Border Layout Example");
        frame.setSize(400, 300);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setLayout(new BorderLayout());

        // Create buttons
        JButton button1 = new JButton("North");
        JButton button2 = new JButton("South");
        JButton button3 = new JButton("East");
        JButton button4 = new JButton("West");
        JButton button5 = new JButton("Center");

        // Add buttons to the frame with Border Layout
```

```
        frame.add(button1, BorderLayout.NORTH);
        frame.add(button2, BorderLayout.SOUTH);
        frame.add(button3, BorderLayout.EAST);
        frame.add(button4, BorderLayout.WEST);
        frame.add(button5, BorderLayout.CENTER);

        // Set frame visibility
        frame.setVisible(true);
    }
}
```

Q2) Define a class CricketPlayer (name,no_of_innings,no_of_times_notout, totatruns, bat_avg). Create an array of n player objects. Calculate the batting average for each player using static method avg(). Define a static sort method which sorts the array on the basis of average. Display the player details in sorted order.

```
import java.io.*;
import java.util.*;
class cricket
{
String name;
int inn,out,runs,avg;
cricket()
{
name = "Dhoni";
inn = 80;
out = 10;
runs = 5000;
}
cricket(String name,int inn,int out,int runs)
{
this.name= name;
this.inn= inn;
this.out=out;
this.runs=runs;
}
static void avgs(cricket c[])
{
for(int i=0;i<c.length;i++)
c[i].avg = c[i].runs / c[i].inn ;
}
static void sort(cricket c[])
{
int i,j;
```

```
cricket c1= new cricket();
for(i=0;i<c.length;i++)
{
for(j=i+1;j<c.length;j++)

{

if(c[i].avg > c[j].avg)
{
c1 = c[i];
c[i]=c[j];
c[j]=c1;

}
}
}
}
void display()
{
System.out.println("Player name is:: "+name+ "\n Player
innings played are" +inn+ " \n Number of times out " +out+"\n Total runs
are " +runs+"\n Batting average is " +this.avg);
}

public static void main(String args[])
{

Scanner sc =new Scanner(System.in);
System.out.println("********** Default
Information*********");
cricket c1 =new cricket();
c1.display();
int n,i;
String name;
int inn,out,runs,avg;
System.out.println("How many player information you
want");
n = sc.nextInt();
cricket c[] = new cricket[n];
for(i =0;i<n;i++)
{
System.out.println("Enter the player name");

name= sc.next();
```

```
System.out.println("Enter the player innings

played");
inn=sc.nextInt();
System.out.println("Enter the player number of
times out");
out=sc.nextInt();

System.out.println("Enter the players total

runs");
runs=sc.nextInt();
c[i]=new cricket(name,inn,out,runs);
}
System.out.println("**********Player
Information*********");
avgs(c);
for(i =0;i<n;i++)
c[i].display();
sort(c);
System.out.println("**********Player Information after
sorting according to batting avg*********");
for(i =0;i<n;i++)
c[i].display();
}
}
```

SLIP 19

Q1) Write a program to accept the two dimensional array from user and display sum of its diagonal elements.

```
import java.util.Scanner;

public class DiagonalSum {

   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);

      // Accepting the size of the array
      System.out.print("Enter the number of rows (and columns for a square matrix): ");
      int n = scanner.nextInt();
```

```
        // Creating a two-dimensional array
        int[][] matrix = new int[n][n];

        // Accepting elements of the matrix
        System.out.println("Enter the elements of the matrix:");
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {
                matrix[i][j] = scanner.nextInt();
            }
        }

        // Calculating the sum of the diagonal elements
        int diagonalSum = 0;
        for (int i = 0; i < n; i++) {
            diagonalSum += matrix[i][i]; // Sum for the primary diagonal
            // Uncomment the next line if you want the sum of the secondary diagonal as well
            // diagonalSum += matrix[i][n - 1 - i]; // Sum for the secondary diagonal
    }
        System.out.println("Sum of diagonal elements: " + diagonalSum);

        scanner.close();
    }
}
```

Q2) Write a program which shows the combo box which includes list of T.Y.B.Sc.(Comp. Sci) subjects. Display the selected subject in a text field.

```
import javax.swing.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class SubjectSelector {

    public static void main(String[] args) {
        // Create the main frame
        JFrame frame = new JFrame("T.Y.B.Sc. (Comp. Sci) Subjects");
        frame.setSize(400, 200);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setLayout(null);

        // Create a combo box with subjects
        String[] subjects = {
            "Data Structures",
```

```
            "Database Management Systems",
            "Operating Systems",
            "Computer Networks",
            "Software Engineering",
            "Web Technologies"
        };

        JComboBox<String> subjectComboBox = new JComboBox<>(subjects);
        subjectComboBox.setBounds(50, 30, 280, 30);

        // Create a text field to display the selected subject
        JTextField selectedSubjectField = new JTextField();
        selectedSubjectField.setBounds(50, 70, 280, 30);
        selectedSubjectField.setEditable(false);

        // Add action listener to the combo box
        subjectComboBox.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                String selectedSubject = (String) subjectComboBox.getSelectedItem();
                selectedSubjectField.setText(selectedSubject);
            }
        });

        // Add components to the frame
        frame.add(subjectComboBox);
        frame.add(selectedSubjectField);

        // Set frame visibility
        frame.setVisible(true);
    }
}
```

SLIP 20

Q1) Write a Program to illustrate multilevel Inheritance such that country is inherited from continent. State is inherited from country. Display the place, state, country and continent.

```
import java.io.*;
import java.util.*;
class continent
{
String cont;
```

```
Scanner sc =new Scanner(System.in);
void cont_input()
{
System.out.println("Enter the continent name");
cont = sc.next();
}
}
class country extends continent
{
String con;
Scanner sc =new Scanner(System.in);
void con_input()
{
System.out.println("Enter the contry name");

con = sc.next();

}
}
class state extends country
{

String sta;
Scanner sc =new Scanner(System.in);
void sta_input()
{
System.out.println("Enter the State name");
sta = sc.next();
}
}
class place extends state
{
String pla;
Scanner sc =new Scanner(System.in);
void pla_input()
{
System.out.println("Enter the Place name");

pla = sc.next();

}
}
class indemo2 extends place
{
```

```
public static void main(String args[])
{
indemo2 p = new indemo2();
p.cont_input();
p.con_input();
p.sta_input();
p.pla_input();

System.out.println("Continent name is :: " + p.cont);
System.out.println("Country name is :: " + p.con);
System.out.println("State name is :: " + p.sta);
System.out.println("Place name is :: " + p.pla);
}
}
```

Q2) Write a package for Operation, which has two classes, Addition and Maximum. Addition has two methods add () and subtract (), which are used to add two integers and subtract two, float values respectively. Maximum has a method max () to display the maximum of two itegers

```
import Operation.Addition;
import Operation.Maximum;

public class Main {
   public static void main(String[] args) {
      Addition addition = new Addition();
      Maximum maximum = new Maximum();

      // Test Addition methods
      int sum = addition.add(10, 5);
      float difference = addition.subtract(10.5f, 4.2f);

      System.out.println("Sum of integers: " + sum);
      System.out.println("Difference of floats: " + difference);

      // Test Maximum method
      int max = maximum.max(10, 20);
      System.out.println("Maximum of two integers: " + max);
   }
}
```

SLIP 21

Q1) Define a class MyDate(Day, Month, year) with methods to accept and display a MyDateobject. Accept date as dd,mm,yyyy. Throw user defined exception "InvalidDateException" if the date is invalid.

```
import java.util.Scanner;

// Custom exception class
class InvalidDateException extends Exception {
    public InvalidDateException(String message) {
        super(message);
    }
}

// MyDate class
public class MyDate {
    private int day;
    private int month;
    private int year;

    // Method to accept date
    public void acceptDate() throws InvalidDateException {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter date (dd mm yyyy): ");
        day = scanner.nextInt();
        month = scanner.nextInt();
        year = scanner.nextInt();

        // Validate the date
        if (!isValidDate(day, month, year)) {
            throw new InvalidDateException("Invalid date: " + day + "/" + month + "/" + year);
        }
    }

    // Method to display date
    public void displayDate() {
        System.out.println("Date: " + day + "/" + month + "/" + year);
    }

    // Method to validate the date
    private boolean isValidDate(int day, int month, int year) {
        if (year < 1 || month < 1 || month > 12 || day < 1) {
            return false;
        }
```

```
        int[] daysInMonth = {31, 28 + (isLeapYear(year) ? 1 : 0), 31, 30, 31, 30, 31, 31, 30, 31,
30, 31};

        return day <= daysInMonth[month - 1];
    }

    // Method to check for leap year
    private boolean isLeapYear(int year) {
        return (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0);
    }

    // Main method
    public static void main(String[] args) {
        MyDate myDate = new MyDate();

        try {
            myDate.acceptDate();
            myDate.displayDate();
        } catch (InvalidDateException e) {
            System.out.println(e.getMessage());
        }
    }
}
```

Q2) Create an employee class(id,name,deptname,salary). Define a default and parameterized constructor. Use ‘this’’ keyword to initialize instance variables. Keep a count of objects created. Create objects using parameterized constructor and display the object count after each object is created. (Use static member and method). Also display the contents of each object.

```
import java.io.*;
import java.util.*;
class emp
{
int eid;
String name,dname;
float salary;
static int cnt =0;

emp()
{
cnt ++;
}
```

```
emp(int eid,String name, String dname, float salary)
{
this.eid= eid;
this.name=name;
this.dname = dname;
this.salary = salary;
cnt++;
}
void display()
{
System.out.println("emp id is " +eid +"\n Employee name is :" +
name + "\n Employee department is " +dname +"\n Employee salary is " +salary);
}
static void count()
{
System.out.println("Number of objects created are " +cnt);
}

public static void main(String args[])
{
int i,n,id;
String name,dname;
float salary;
Scanner sc =new Scanner(System.in);
System.out.println("****** Default constructor values are**********");
emp e1 = new emp();
e1.display();
System.out.println("How many employee you want");
n = sc.nextInt();
emp e[] = new emp[n];
for(i=0;i<n;i++)
{
System.out.println("Enter the id");
id = sc.nextInt();
System.out.println("Enter the name");
name = sc.next();
System.out.println("Enter the department name");
dname = sc.next();
System.out.println("Enter the salary");
salary = sc.nextFloat();

e[i] = new emp(id,name,dname,salary);
}
for(i=0;i<n;i++)
```

```
e[i].display();
e1.count();
}
}
```

SLIP 22

Q1) Write a program to create an abstract class named Shape that contains two integers and an empty method named printArea(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contain only the method printArea() that prints the area of the given shape. (use method overriding).

```
import java.io.*;
import java.util.*;
abstract class shape
{
int n1,n2;
public abstract void printarea();
}
class circle extends shape
{
circle(int a)
{
super.n1= a;
}

public void printarea()
{
System.out.println("Area of cirlce is::

"+(3.14*super.n1*super.n1));
}
}

class rectangle extends shape
{
rectangle(int l,int b)
{
super.n1= l;
super.n2 = b;

}
public void printarea()
```

```
{
System.out.println("Area of Rectangle is:: "+(super.n1*super.n2));
}
}
class triangle extends shape
{
triangle(int h,int b)
{
super.n1= h;
super.n2 = b;
}
public void printarea()
{
System.out.println("Area of Triangle is:: "+(super.n1*super.n2)/2);
}
}
class sdemo
{
public static void main(String args[])
{
circle c = new circle(2);
rectangle r=new rectangle(4,5);

triangle t=new triangle(3,2);
c.printarea();
r.printarea();
t.printarea();
}
}
```

Q2) Write a program that handles all mouse events and shows the event name at the center of the Window, red in color when a mouse event is fired. (Use adapter classes).

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;

public class MouseEventExample extends JFrame {
   private JLabel eventLabel;

   public MouseEventExample() {
      // Set up the frame
      setTitle("Mouse Event Example");
```

```
    setSize(400, 300);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setLayout(new BorderLayout());

    // Create a label to display the event name
    eventLabel = new JLabel("Mouse Event", SwingConstants.CENTER);
    eventLabel.setFont(new Font("Arial", Font.BOLD, 24));
    eventLabel.setForeground(Color.RED);
    add(eventLabel, BorderLayout.CENTER);

    // Add mouse listener using MouseAdapter
    addMouseListener(new MouseAdapter() {
        @Override
        public void mouseClicked(MouseEvent e) {
            eventLabel.setText("Mouse Clicked");
        }

        @Override
        public void mousePressed(MouseEvent e) {
            eventLabel.setText("Mouse Pressed");
        }

        @Override
        public void mouseReleased(MouseEvent e) {
            eventLabel.setText("Mouse Released");
        }

        @Override
        public void mouseEntered(MouseEvent e) {
            eventLabel.setText("Mouse Entered");
        }

        @Override
        public void mouseExited(MouseEvent e) {
            eventLabel.setText("Mouse Exited");
        }
    });
}

public static void main(String[] args) {
    SwingUtilities.invokeLater(() -> {
        MouseEventExample example = new MouseEventExample();
        example.setVisible(true);
    });
```

```
   }
}
```

SLIP 23

Q1) Define a class MyNumber having one private int data member. Write a default constructor to initialize it to 0 and another constructor to initialize it to a value (Use this). Write methods isNegative, isPositive, isZero, isOdd, isEven. Create an object in main.Use command line arguments to pass a value to the Object.

```
import java.util.*;
class mynumber
{
private int n;
mynumber()

{
n = 0;
}
mynumber(int n)
{
this.n= n;
}
public void ispositive(int x)
{
if(x>0)
System.out.println("Positive");
}
public void isnegative(int x)
{
if(x<0)
System.out.println("Negative");
}

public void iseven(int x)
{
if(x%2==0)
System.out.println("Even");
}
public void isodd(int x)
{
if(x%2==1)
System.out.println("Odd");
}
```

```
public static void main(String args[])
{
int n = Integer.parseInt(args[0]);
mynumber m = new mynumber();
m.ispositive(n);
m.isnegative(n);
m.iseven(n);
m.isodd(n);
}
}
```

Q2) Write a simple currency converter, as shown in the figure. User can enter the amount of "Singapore Dollars", "US Dollars", or "Euros", in floating-point number. The converted values shall be displayed to 2 decimal places. Assume that 1 USD = 1.41 SGD, 1 USD = 0.92 Euro, 1 SGD = (.65 Euro.

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class CurrencyConverter extends JFrame {
   private JTextField amountField;
   private JTextArea resultArea;

   // Conversion rates
   private static final double USD_TO_SGD = 1.41;
   private static final double USD_TO_EURO = 0.92;
   private static final double SGD_TO_EURO = 0.65;

   public CurrencyConverter() {
      setTitle("Currency Converter");
      setSize(400, 200);
      setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      setLayout(new FlowLayout());

      amountField = new JTextField(10);
      String[] currencies = {"SGD", "USD", "EUR"};
      JComboBox<String> currencySelect = new JComboBox<>(currencies);
      JButton convertButton = new JButton("Convert");

      resultArea = new JTextArea(5, 30);
      resultArea.setEditable(false);
```

```
        convertButton.addActionListener(e ->
convert(currencySelect.getSelectedItem().toString()));

        add(new JLabel("Amount:"), BorderLayout.NORTH);
        add(amountField);
        add(currencySelect);
        add(convertButton);
        add(new JScrollPane(resultArea));
    }

    private void convert(String currency) {
        double amount;
        try {
            amount = Double.parseDouble(amountField.getText());
        } catch (NumberFormatException ex) {
            resultArea.setText("Invalid number.");
            return;
        }

        StringBuilder result = new StringBuilder();
        switch (currency) {
            case "SGD":
                result.append("USD: ").append(String.format("%.2f", amount /
USD_TO_SGD)).append("\n");
                result.append("EUR: ").append(String.format("%.2f", amount *
SGD_TO_EURO)).append("\n");
                break;
            case "USD":
                result.append("SGD: ").append(String.format("%.2f", amount *
USD_TO_SGD)).append("\n");
                result.append("EUR: ").append(String.format("%.2f", amount *
USD_TO_EURO)).append("\n");
                break;
            case "EUR":
                result.append("SGD: ").append(String.format("%.2f", amount /
SGD_TO_EURO)).append("\n");
                result.append("USD: ").append(String.format("%.2f", amount /
USD_TO_EURO)).append("\n");
                break;
        }
        resultArea.setText(result.toString());
    }

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

```
        SwingUtilities.invokeLater(() -> {
            CurrencyConverter converter = new CurrencyConverter();
            converter.setVisible(true);
        });
    }
}
```

SLIP 24

Q1) Create an abstract class 'Bank' with an abstract method 'getBalance'. Rs.100, Rs.150 and Rs.200 are deposited in banks A, B and C respectively. 'BankA', 'BankB' and 'BankC' are subclasses of class 'Bank', each having a method named 'getBalance'. Call this method by creating an object of each of the three classes.

```
import java.io.*;
abstract class bank
{
public abstract void getbalance();
}
class bankA extends bank
{
double balance = 100;
public void getbalance()
{
System.out.println("Bank balance of BankA is "+balance);
}
}
class bankB extends bank
{
double balance = 150;
public void getbalance()
{
System.out.println("Bank balance of BankB is "+balance);
}
}
class bankC extends bank
{
double balance = 200;
public void getbalance()
{
System.out.println("Bank balance of BankC is "+balance);
}
}
```

```
class bankdemo
{
public static void main(String args[])
{
bankA ba = new bankA();
bankB bb = new bankB();
bankC bc = new bankC();
ba.getbalance();
bb.getbalance();
bc.getbalance();
}
}
```

Q2) Program that displays three concentric circles where ever the user clicks the mouse on a frame. The program must exit when user clicks “X” on the frame.

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;

public class ConcentricCircles extends JFrame {
   private int x, y;

   public ConcentricCircles() {
      setTitle("Concentric Circles");
      setSize(400, 400);
      setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      addMouseListener(new MouseAdapter() {
         @Override
         public void mouseClicked(MouseEvent e) {
            x = e.getX();
            y = e.getY();
            repaint(); // Trigger a repaint to draw circles
         }
      });
   }

   @Override
   public void paint(Graphics g) {
      super.paint(g);
      g.setColor(Color.BLUE);
      g.drawOval(x - 50, y - 50, 100, 100); // Outer circle
      g.setColor(Color.GREEN);
```

```
        g.drawOval(x - 30, y - 30, 60, 60);   // Middle circle
        g.setColor(Color.RED);
        g.drawOval(x - 10, y - 10, 20, 20);    // Inner circle
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> {
            ConcentricCircles frame = new ConcentricCircles();
            frame.setVisible(true);
        });
    }
}
```

SLIP 25

Q1) Create a class Student(rollno, name ,class, per), to read student information from the console and display them (Using BufferedReader class)

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

class Student {
    private int rollno;
    private String name, className;
    private double percentage;

    public Student(int rollno, String name, String className, double percentage) {
        this.rollno = rollno; this.name = name; this.className = className; this.percentage
= percentage;
    }

    public void display() {
        System.out.printf("Roll No: %d\nName: %s\nClass: %s\nPercentage: %.2f%%\n",
rollno, name, className, percentage);
    }
}

public class StudentInfo {
    public static void main(String[] args) {
        try (BufferedReader reader = new BufferedReader(new
InputStreamReader(System.in))) {
            System.out.print("Enter Roll No: ");
```

```
            int rollno = Integer.parseInt(reader.readLine());
            System.out.print("Enter Name: ");
            String name = reader.readLine();
            System.out.print("Enter Class: ");
            String className = reader.readLine();
            System.out.print("Enter Percentage: ");
            double percentage = Double.parseDouble(reader.readLine());
            new Student(rollno, name, className, percentage).display();
        } catch (IOException | NumberFormatException e) {
            System.out.println("Invalid input: " + e.getMessage());
        }
    }
}
```

Q2) Create the following GUI screen using appropriate layout manager. Accept the name, class, hobbies from the user and display the selected options in a textbox.

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;

public class UserInfoForm {
    public static void main(String[] args) {
        JFrame frame = new JFrame("User Info Form");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(400, 300);

        JPanel panel = new JPanel(new GridLayout(6, 2));

        // Your Name
        panel.add(new JLabel("Your Name"));
        JTextField nameField = new JTextField();
        panel.add(nameField);

        // Your Class
        panel.add(new JLabel("Your Class"));
        ButtonGroup classGroup = new ButtonGroup();
        JRadioButton[] classButtons = { new JRadioButton("FY"), new JRadioButton("SY"),
new JRadioButton("TY") };
        for (JRadioButton button : classButtons) {
            classGroup.add(button);
        }
        JPanel classPanel = new JPanel(new GridLayout(3, 1));
        for (JRadioButton button : classButtons) classPanel.add(button);
```

```
        panel.add(classPanel);

        // Your Hobbies
        panel.add(new JLabel("Your Hobbies"));
        JCheckBox[] hobbyBoxes = { new JCheckBox("Music"), new JCheckBox("Dance"),
new JCheckBox("Sports") };
        JPanel hobbiesPanel = new JPanel(new GridLayout(3, 1));
        for (JCheckBox box : hobbyBoxes) hobbiesPanel.add(box);
        panel.add(hobbiesPanel);

        // Output TextField
        JTextField outputField = new JTextField();
        outputField.setEditable(false);

        // Submit Button
        JButton submitButton = new JButton("Submit");
        submitButton.addActionListener((ActionEvent e) -> {
            String name = nameField.getText();
            String selectedClass = classButtons[0].isSelected() ? "FY" :
                                classButtons[1].isSelected() ? "SY" : "TY";
            String hobbies = "";
            for (JCheckBox box : hobbyBoxes) if (box.isSelected()) hobbies += box.getText() +
" ";
            outputField.setText("Name: " + name + " | Class: " + selectedClass + " | Hobbies: "
+ hobbies);
        });

        panel.add(submitButton);
        panel.add(outputField);

        frame.add(panel);
        frame.setVisible(true);
    }
}
```

SLIP 26

Q1) Define a Item class (item_number, item_name, item_price). Define a default and parameterized constructor. Keep a count of objects created. Create objects using parameterized constructor and display the object count after each object is created.(Use static member and method). Also display the contents of each object.

```
import java.io.*;
```

```
import java.util.*;
class item
{
int ino;
String name,dname;
float price;
static int cnt = 0;
item()
{
cnt++;
}
item(int ino, String name,String dname,float price)
{
this.ino = ino;
this.name=name;
this.dname = dname;
this.price = price;
cnt++;
}
void display()
{
System.out.println("Id is = " +this.ino + "\n Item name = " +
this.name +"\n Department name= "+this.dname +"\n Price = "+this.price + "\n number
of object created =" +cnt);
}
public static void main(String args[])
{
item i= new item();
i.display();
item i1= new item(123,"Pen","Stationary",100);
i1.display();
}
}
```

Q2) Define a class ‘Donor’ to store the below mentioned details of a blood donor. name, age, address, contactnumber, bloodgroup, date of last donation. Create ‘n’ objects of this class for all the regular donors at Pune. Write these objects to a file. Read these objects from the file and display only those donors’’ details whose blood group 1s “A+ve’ and had not donated for the recent six months.

```
import java.io.*;
import java.time.LocalDate;
import java.util.ArrayList;
import java.util.List;
```

```
class Donor implements Serializable {
   private String name, address, contactNumber, bloodGroup;
   private int age;
   private LocalDate lastDonationDate;

   public Donor(String name, int age, String address, String contactNumber, String
bloodGroup, LocalDate lastDonationDate) {
      this.name = name; this.age = age; this.address = address;
      this.contactNumber = contactNumber; this.bloodGroup = bloodGroup;
this.lastDonationDate = lastDonationDate;
   }

   public String getBloodGroup() { return bloodGroup; }
   public LocalDate getLastDonationDate() { return lastDonationDate; }

   @Override
   public String toString() {
      return String.format("Name: %s, Age: %d, Address: %s, Contact: %s, Blood Group:
%s, Last Donation: %s",
                           name, age, address, contactNumber, bloodGroup, lastDonationDate);
   }
}

public class DonorManager {
   private static final String FILE_NAME = "donors.dat";

   public static void main(String[] args) {
      List<Donor> donors = List.of(
         new Donor("John Doe", 30, "123 Main St", "9876543210", "A+ve",
LocalDate.now().minusMonths(7)),
         new Donor("Jane Smith", 28, "456 Elm St", "9123456780", "B+ve",
LocalDate.now().minusMonths(4)),
         new Donor("Alice Johnson", 35, "789 Oak St", "9988776655", "A+ve",
LocalDate.now().minusMonths(8))
      );

      writeDonorsToFile(donors);
      readAndDisplayEligibleDonors();
   }

   private static void writeDonorsToFile(List<Donor> donors) {
      try (ObjectOutputStream oos = new ObjectOutputStream(new
FileOutputStream(FILE_NAME))) {
```

```
            oos.writeObject(donors);
        } catch (IOException e) {
            System.err.println("Error writing to file: " + e.getMessage());
        }
    }

    private static void readAndDisplayEligibleDonors() {
        try (ObjectInputStream ois = new ObjectInputStream(new
FileInputStream(FILE_NAME))) {
            List<Donor> donors = (List<Donor>) ois.readObject();
            LocalDate sixMonthsAgo = LocalDate.now().minusMonths(6);
            donors.stream()
                .filter(d -> d.getBloodGroup().equals("A+ve") &&
d.getLastDonationDate().isBefore(sixMonthsAgo))
                .forEach(System.out::println);
        } catch (IOException | ClassNotFoundException e) {
            System.err.println("Error reading from file: " + e.getMessage());
        }
    }
}
```

SLIP 27

Q1) Define an Employee class with suitable attributes having getSalary() method, which returns salary withdrawn by a particular employee. Write a class Manager which extends a class Employee, override the getSalary() method, which will return salary of manager by adding traveling allowance, house rent allowance etc.

```
import java.io.*;
import java.util.*;
class employee
{
double salary;
employee()
{
salary = 35000;
}
public double getsalary()
{
Scanner sc = new Scanner(System.in);

double withdraw;
System.out.println("Enter the salary which employee want to
```

```
withdraw ");

withdraw =sc.nextDouble();
return (salary - withdraw);
}
}
class manager extends employee
{
double traveling,rent;
manager()
{
traveling = 2000;
rent= 5000;
}
public double getsalary()
{
return (super.salary + traveling+rent);
}
}
class edemo
{
public static void main(String args[])
{
employee e = new employee();
manager m = new manager();
System.out.println("Employee salary is " + e.salary +

"\nEmployee salary ifter withdraw is" +e.getsalary());

System.out.println("Manager Salary is"+m.getsalary());
}
}
```

Q2) Write a program to accept a string as command line argument and check whether it is a file or directory. Also perform operations as follows: plf it is a directory,delete all text files in that directory. Confirm delete operation from user before deleting text files. Also, display a count showing the number of files deleted, if any, from the directory. i)If it is a file display various details of that file.

SLIP 28

Q1) Write a program that reads on file name from the user, then displays information about whether the file exists, whether the file is readable, whether the file is writable, the type of file and the length of the file in bytes.

```
import java.io.File;
import java.util.Scanner;

public class FileInfoReader {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
      System.out.print("Enter the file name (with path if not in current directory): ");
      String fileName = scanner.nextLine();

      File file = new File(fileName);

      if (file.exists()) {
         System.out.println("File exists: " + file.exists());
         System.out.println("Readable: " + file.canRead());
         System.out.println("Writable: " + file.canWrite());
         System.out.println("File type: " + (file.isDirectory() ? "Directory" : "File"));
         System.out.println("File length: " + file.length() + " bytes");
      } else {
         System.out.println("The file does not exist.");
      }

      scanner.close();
   }
}
```

Q2) Write a program called SwingTemperatureConverter to convert temperature values between Celsius and Fahrenheit. User can enter either the Celsius or the Fahrenheit value, in floating-point number. Hints: To display a floating-point number in a specific format (e.g., 1 decimal place), use the static method String.format(), which has the same form as printf(). For example, String.format("%.1f", 1.234) returns String "1.2".

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;

public class SwingTemperatureConverter {
   public static void main(String[] args) {
      JFrame frame = new JFrame("Temperature Converter");
      frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      frame.setSize(300, 200);

      JPanel panel = new JPanel(new GridLayout(3, 2));
```

```
        JLabel celsiusLabel = new JLabel("Celsius:");
        JTextField celsiusField = new JTextField();
        JLabel fahrenheitLabel = new JLabel("Fahrenheit:");
        JTextField fahrenheitField = new JTextField();

        JButton convertToF = new JButton("Convert to Fahrenheit");
        JButton convertToC = new JButton("Convert to Celsius");

        convertToF.addActionListener((ActionEvent e) -> {
            try {
                double celsius = Double.parseDouble(celsiusField.getText());
                double fahrenheit = celsius * 9 / 5 + 32;
                fahrenheitField.setText(String.format("%.1f", fahrenheit));
            } catch (NumberFormatException ex) {
                JOptionPane.showMessageDialog(frame, "Invalid Celsius value");
            }
        });

        convertToC.addActionListener((ActionEvent e) -> {
            try {
                double fahrenheit = Double.parseDouble(fahrenheitField.getText());
                double celsius = (fahrenheit - 32) * 5 / 9;
                celsiusField.setText(String.format("%.1f", celsius));
            } catch (NumberFormatException ex) {
                JOptionPane.showMessageDialog(frame, "Invalid Fahrenheit value");
            }
        });

        panel.add(celsiusLabel);
        panel.add(celsiusField);
        panel.add(convertToF);
        panel.add(fahrenheitLabel);
        panel.add(fahrenheitField);
        panel.add(convertToC);

        frame.add(panel);
        frame.setVisible(true);
    }
}
```

SLIP 29

Q1) Write a program to create a class Customer(custno,custname,contactnumber,custaddr). Write a method to search the customer name with given contact number and display the details.

```
import java.io.*;
import java.util.*;
class customer

{
int cid,n,i;
String name,cno,address;
static void search(String cno, customer a[])
{
int i,flag = 0;
for(i=0;i<a.length;i++)
{
if(cno.equals(a[i].cno))
{
a[i].display();
flag = 1;
}
}
if (flag == 0)
System.out.println("Record not found");
}
void display()
{
System.out.println("cid = " + cid +"\n Name is = "+name +"\n Adress
is ="+address+"\n Contact number is =" +cno);
}
public static void main(String args[])
{
int n,i;
String number;
Scanner sc =new Scanner (System.in);
System.out.println("How many customer you want");
n = sc.nextInt();
customer c[]=new customer[n];
for(i=0;i<n;i++)
{
c[i]=new customer();
System.out.println("Enter the cid");
c[i].cid = sc.nextInt();
System.out.println("Enter the name");
```

```
c[i].name = sc.next();
System.out.println("Enter the address");
c[i].address = sc.next();
System.out.println("Enter the contact number");
c[i].cno = sc.next();
}
System.out.println("********* Customer
Information**************");
for(i=0;i<n;i++)
c[i].display();
System.out.println("Enter the number which you want to serach");

number = sc.next();
search(number,c);
}
}
```

Q2) Write a program to create a super class Vehicle having members Company and price. Derive two different classes LightMotorVehicle(mileage) and HeavyMotorVehicle (capacity_in_tons). Accept the information for "n" vehicles and display the information in appropriate form. While taking data, ask user about the type of vehicle first.

```
import java.io.*;
import java.util.*;
class vehicle
{
String cname;
double price;
public void accept()
{
Scanner sc = new Scanner(System.in);
System.out.println("Enter the company name");
cname = sc.next();
System.out.println("Enter the vehicle price");
price = sc.nextDouble();
}
public void display()
{
System.out.println("Company name is " +cname+ "\n vehicle

price is " +price);
}
}
```

```
class LightMotorVehicle extends vehicle
{
double milage;
public void accept()
{
super.accept();
Scanner sc = new Scanner(System.in);
System.out.println("Enter the vehicle milage");
milage = sc.nextDouble();
}
public void display()
{
super.display();
System.out.println("Vehicle milage is " +milage);
}
}
class HeavyMotorVehicle extends vehicle
{
double capacity;
public void accept()
{
super.accept();
Scanner sc = new Scanner(System.in);
System.out.println("Enter the vehicle capacity");
capacity = sc.nextDouble();
}
public void display()
{
super.display();
System.out.println("Vehicle capacity is " +capacity);
}
}
class vdemo
{
public static void main(String args[])
{
int i,ch,n;
Scanner sc =new Scanner(System.in);
System.out.println(" Enter the type of vehicle you want:: \n

1.LightMotorVehicle \n 2. HeavyMotorVehicle");

ch = sc.nextInt();
switch(ch)
```

```
{
case 1: System.out.println("How many vehicle you

want");

n = sc.nextInt();
LightMotorVehicle l[] =new LightMotorVehicle[n];
for(i=0;i<n;i++)
{
l[i]=new LightMotorVehicle();

l[i].accept();
}
System.out.println("******Light Motor Vehicle

Informatiom ***********");

for(i=0;i<n;i++)
l[i].display();
break;

case 2: System.out.println("How many vehicle you
want");

n = sc.nextInt();
HeavyMotorVehicle h[] =new HeavyMotorVehicle[n];

for(i=0;i<n;i++)

{

h[i]=new HeavyMotorVehicle();

h[i].accept();

}
System.out.println("******Heavy Motor
Vehicle Informatiom ***********");
for(i=0;i<n;i++)
h[i].display();

break;
default : System.out.println("Enter proper choice");
}
```

```
}
}
```

SLIP 30

Q1) Write program to define class Person with data member as Personname, Aadharno, Panno. Accept information for 5 objects and display appropriate information (use this keyword).

```
import java.util.Scanner;

class Person {
   private String name, aadhar, pan;

   public Person(String name, String aadhar, String pan) {
      this.name = name; this.aadhar = aadhar; this.pan = pan;
   }

   public void display() {
      System.out.printf("Name: %s\nAadhar: %s\nPAN: %s\n----------------------------\n",
name, aadhar, pan);
   }
}

public class PersonInfo {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
      Person[] persons = new Person[5];

      for (int i = 0; i < 5; i++) {
         System.out.printf("Enter details for Person %d:\n", i + 1);
         System.out.print("Name: ");
         String name = scanner.nextLine();
         System.out.print("Aadhar: ");
         String aadhar = scanner.nextLine();
         System.out.print("PAN: ");
         String pan = scanner.nextLine();
         persons[i] = new Person(name, aadhar, pan);
      }

      System.out.println("\nPerson Information:");
      for (Person person : persons) person.display();
```

```
        scanner.close();
    }
}
```

Q2) Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Numberl and Number2. The division of Numberl and Number2 is displayed in the Result field when the Divide button is clicked. If Number] or Number2 were not an integer, the program would throw a NumberFormatException. If Number2 were Zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box.

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;

public class IntegerDivisionApp {
    public static void main(String[] args) {
        JFrame frame = new JFrame("Integer Division");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(300, 200);

        JPanel panel = new JPanel(new GridLayout(4, 2));

        JLabel num1Label = new JLabel("Number 1:");
        JTextField num1Field = new JTextField();

        JLabel num2Label = new JLabel("Number 2:");
        JTextField num2Field = new JTextField();

        JButton divideButton = new JButton("Divide");
        JTextField resultField = new JTextField();
        resultField.setEditable(false);

        divideButton.addActionListener((ActionEvent e) -> {
            try {
                int num1 = Integer.parseInt(num1Field.getText());
                int num2 = Integer.parseInt(num2Field.getText());
                int result = num1 / num2;
                resultField.setText(String.valueOf(result));
            } catch (NumberFormatException ex) {
                JOptionPane.showMessageDialog(frame, "Invalid input. Please enter integers.",
"Input Error", JOptionPane.ERROR_MESSAGE);
            } catch (ArithmeticException ex) {
```

```
            JOptionPane.showMessageDialog(frame, "Division by zero is not allowed.",
"Math Error", JOptionPane.ERROR_MESSAGE);
        }
    });

    panel.add(num1Label);
    panel.add(num1Field);
    panel.add(num2Label);
    panel.add(num2Field);
    panel.add(divideButton);
    panel.add(resultField);

    frame.add(panel);
    frame.setVisible(true);
  }
}
```