

SLIP 1

Q1) Write a Program to print all Prime numbers in an array of 'n' elements. (use command line arguments)

SOLUTION:

```
class Main {
    public static int isPrime(int x)
    {
        int i;
        for (i = 2; i < x / 2 + 1; i++) {
            if (x % i == 0) {
                return 0;
            }
        }
        return 1;
    }
    public static void findPrimes(int n)
    {
        int i;
        for (i = 2; i <= n; i++) {
            if (isPrime(i) == 1)
                System.out.print(i + ", ");
        }
        System.out.println();
    }
    public static void main(String[] args)
    {
        if (args.length > 0) {
            int n = Integer.parseInt(args[0]);
            findPrimes(n);
        }
        else
            System.out.println("No command line " + "arguments found.");
    }
}
```

Q2) 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.

SOLUTION:

```
import java.io.IOException;
```

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

```
abstract class Staff{
```

```
String name;
```

```
int id;
```

```
}
```

```
class FullTimeStaff extends Staff{
```

```
String department;
```

```
double salary;
```

```
public void accept() throws IOException{
```

```
System.out.println("Enter the name, ID, department and salary: ");
```

```
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
```

```
name=br.readLine();
```

```
id=Integer.parseInt(br.readLine());
```

```
department=br.readLine();
```

```
salary=Double.parseDouble(br.readLine());
```

```
}
```

```
public void display(){
```

```
System.out.println("Name: "+name);
```

```
System.out.println("ID: "+id);
```

```
System.out.println("Department: "+department);
```

```
System.out.println("Salary: "+salary);
System.out.println("-----");
}
}

class PartTimeStaff extends Staff{
int hours, rate;
public void accept() throws IOException{
System.out.println("Enter the name, ID , No of working hours and rate per hour:");
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
name=br.readLine();
id=Integer.parseInt(br.readLine());
hours=Integer.parseInt(br.readLine());
rate=Integer.parseInt(br.readLine());
}
public void display(){
System.out.println("Name: "+name);
System.out.println("ID: "+id);
System.out.println("No of Working Hours: "+hours);
System.out.println("Rate per hour: "+rate);
System.out.println("-----");
}
}

public class Main {
public static void main(String [] args) throws IOException{
int i;
```

```
System.out.println("Select Any One: ");
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
System.out.println("1.Full Time Staff");
System.out.println("2.Part Time Staff");
int ch=Integer.parseInt(br.readLine());
switch(ch){
case 1:
System.out.println("Enter the number of Full Time Staff: ");
int n=Integer.parseInt(br.readLine());
FullTimeStaff [] l=new FullTimeStaff[n];
for(i=0;i<n;i++){
l[i]=new FullTimeStaff();
l[i].accept();
}
for(i=0;i<n;i++){
l[i].display();
}
break;
case 2:
System.out.println("Enter the number of Part Time Staff: ");
int m=Integer.parseInt(br.readLine());
PartTimeStaff [] h=new PartTimeStaff[m];
for(i=0;i<m;i++){
h[i]=new PartTimeStaff();
```

```
h[i].accept();  
}  
for(i=0;i<m;i++){  
h[i].display();  
}  
break;  
}  
}  
}
```

SLIP 2

Q1) 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 : $BMI = Wts. \text{ In kgs} / (ht)^2$)

SOLUTION:

```
import java.util.Scanner;
public class Main {

    public static void main(String[] Strings) {

        Scanner input = new Scanner(System.in);

        System.out.print("Input First name: ");
        String Fname = input.nextLine();

        System.out.print("Input last name: ");
        String Lname = input.nextLine();

        System.out.print("Input weight in KG: ");
        double weight = input.nextDouble();

        System.out.print("Input height in Meters: ");
        double height = input.nextDouble();

        double BMI = weight / (height * height);
        System.out.print("Body Mass Index is " + BMI + "\n");
    }
}
```

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.

SOLUTION:

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

```
class CricketPlayer {
```

```
    String name;
    int no_of_innings;
    int no_of_times_not_out;
    int totlaruns;
    float bat_avg;
    static int count;
```

```
    CricketPlayer() {
    }
}
```

```
    CricketPlayer(String name, int no_of_innings, int no_of_times_not_out, int
totlaruns , float bat_avg) {
```

```
        count++;
        this.name=name;
        this.no_of_innings=no_of_innings;
        this.no_of_times_not_out=no_of_times_not_out;
        this.totlaruns=totlaruns;
```

```

        this.bat_avg=bat_avg;

    }

    void display() {

        System.out.println("NAME: "+name+" INNINGS: "+no_of_innings+" NOT
OUT: "+no_of_times_not_out+" TOTAL RUNS: "+totlaruns+" BATTING
AVERAGE: "+bat_avg);

    }

    float getBatAvg() {

        return bat_avg;

    }

    void counter() {

        System.out.println(count + " object is created");

    }

    public static void sortStudent(CricketPlayer s[], int n) {

        for (int i = n - 1; i >= 0; i--) {

            for (int j = 0; j < i; j++) {

                if (s[j].getBatAvg() > s[j + 1].getBatAvg()) {

                    CricketPlayer t = s[j];

                    s[j] = s[j + 1];

                    s[j + 1] = t;

                }

            }

        }

        System.out.println("-----List of Player sorted as per Batting Average-----");

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

            s[i].display();

    }

```



```
}
```

```
public class ass2_SetB_q2 {  
    public static void main(String args[]) throws IOException {  
        BufferedReader br = new BufferedReader(new  
InputStreamReader(System.in));  
        System.out.println("Enter no. of Player:");  
        int n = Integer.parseInt(br.readLine());  
        CricketPlayer p[] = new CricketPlayer[n];  
        for (int i = 0; i < n; i++) {  
            System.out.print("Enter Name:");  
            String name = br.readLine();  
            System.out.print("Enter Innings:");  
            int no_of_innings = Integer.parseInt(br.readLine());  
            System.out.println("Enter number of times not out");  
            int no_of_times_not_out = Integer.parseInt(br.readLine());  
            System.out.println("Enter total runs");  
            int totlaruns = Integer.parseInt(br.readLine());  
            System.out.println("Enter Batting Average");  
            float bat_avg = Float.parseFloat(br.readLine());  
  
            p[i] = new  
CricketPlayer(name,no_of_innings,no_of_times_not_out,totlaruns,bat_avg);  
            p[i].counter();  
        }  
        CricketPlayer.sortStudent(p, CricketPlayer.count);  
    }  
}
```

SLIP 3

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

SOLUTION:

```
class Main
```

```
{  
    public static void main(String arg[])  
    {  
        String name[]=new String[10];  
        int l=arg.length;  
        String temp;  
        for(int i=0;i<l;i++)  
        {  
            name[i]=arg[i];  
        }  
        for(int j=0;j<l;j++)  
        {  
            for(int k=j+1;k<l;k++)  
            {  
                if((name[j].compareTo(name[k]))>0)  
                {  
                    temp=name[j];  
                    name[j]=name[k];  
                    name[k]=temp;  
                }  
            }  
        }  
    }  
}
```

```

    }
}
System.out.println("Sorted City Are-");
for(int i=0;i<l;i++)
{
    System.out.println(name[i]);
}
}
}

```

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 is Covid Positive(+)" and Need to Hospitalized" otherwise display its information.

SOLUTION:

```

import java.util.*;

class patient
{
    String name;
    int age;
    int oxylevel;
    int HRCTreport;

    patient(String name, int age, int oxylevel, int HRCTreport)
    {
        this.name = name;
        this.age = age;

```

```
this.oxylevel = oxylevel;
this.HRCTreport = HRCTreport;
}
}

public class Main extends Exception
{
public static void main(String[] args)
{
Scanner sc = new Scanner(System.in);
System.out.println("How many patient you want insert:");
int number = sc.nextInt();
patient[] ob = new patient[number];
for(int j=0; j<number; j++)
{
System.out.println("Enter Name ");
String name = sc.next();
System.out.println("Enter Age ");
int age = sc.nextInt();
System.out.println("Enter oxygen level");
int oxylevel = sc.nextInt();
System.out.println("Enter HRCT report");
int HRCTreport = sc.nextInt();
ob[j] = new patient(name, age, oxylevel, HRCTreport);
}
}
```

```
for(int j=0; j<number; j++)
{
if(ob[j].oxylevel < 95 && ob[j].HRCTreport > 10)
try
{

throw new NullPointerException("\n");
}
catch(Exception e)
{
System.out.println("Patient is Covid Positive(+) and Need to Hospitalized\n");

}
else
{
System.out.println("name: "+ob[j].name);
System.out.println("age " + ob[j].age);
System.out.println("oxygen level " +ob[j].oxylevel);
System.out.println("HRCT report " + ob[j].HRCTreport);
System.out.println("\n");
}
}
}
}
```

SLIP 4

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

SOLUTION:

```
import java.io.*;

public class Main {
    static BufferedReader sc = new BufferedReader(new
InputStreamReader(System.in));

    static int a[][] = new int[5][5];

    public static void main(String[] args) throws NumberFormatException,
IOException {
        int m = 0, n = 0;

        System.out.print("Enter number of rows ");
        m = Integer.parseInt(sc.readLine());
        System.out.print("Enter number of columns ");
        n = Integer.parseInt(sc.readLine());
        if (m == n)
        {
            int i, j;
            System.out.println("Enter values for matrix: ");
            for (i = 0; i < m; i++) {
                for (j = 0; j < n; j++) {
                    a[i][j] = Integer.parseInt(sc.readLine());
                }
            }
        }
    }
}
```

```
System.out.println("Original Matrix :");
```

```
    for (i = 0; i < m; i++) {
```

```
        for (j = 0; j < n; j++) {
```

```
            System.out.print "[" + a[i][j] + "];
```

```
        }
```

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

```
    }
```

```
int transpose[][] = new int[m][n];
```

```
System.out.println("Transpose of Matrix = ");
```

```
int i3, j3;
```

```
for (i3 = 0; i3 < m; i3++)
```

```
{
```

```
    for (j3 = 0; j3 < n; j3++)
```

```
    {
```

```
        transpose[i3][j3] = a[j3][i3];
```

```
        System.out.print "[" + transpose[i3][j3] + "];
```

```
    }
```

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

```
    }
```

```
}
```

```
else
```

```
{
```

```
    System.out.println("Rows not equal to columns");
```

```
}
```

```
}
```

```
}
```

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.

SOLUTION:

```
import java.awt.*;
import java.awt.event.*;
class InvalidPasswordException extends Exception
{
    InvalidPasswordException()
    {
        System.out.println(" User name and Password is not same");
    }
}
public class PasswordDemo extends Frame implements ActionListener
{
    Label uname,upass;
    TextField nametext;
    TextField passtext,msg;
    Button login,Clear;
    Panel p;
    int attempt=0;
    char c= ' * ' ;

    public void login()
    {
        p=new Panel();
        uname=new Label("Use Name: " ,Label.CENTER);
        upass=new Label ("Password: ",Label.RIGHT);

        nametext=new TextField(20);
        passtext =new TextField(20);
        passtext.setEchoChar(c);
```



```
msg=new TextField(10);
msg.setEditable(false);

login=new Button("Login");
Clear=new Button("Clear");
login.addActionListener(this);
Clear.addActionListener(this);

p.add(uname);
p.add(nametext);
p.add(upass);
p.add(passtext);
p.add(login);
p.add(Clear);
p.add(msg);
add(p);

setTitle("Login ");
setSize(290,200);
setResizable(false);
setVisible(true);
}

public void actionPerformed(ActionEvent ae)
{
    Button btn=(Button)(ae.getSource());
    if(attempt<3)
    {
        if((btn.getLabel()=="Clear"))
        {
            nametext.setText("");
            passtext.setText("");
        }
        if((btn.getLabel()).equals("Login"))
        {
            try
            {
```

```
String user=nametext.getText();
String upass=passtext.getText();

if(user.compareTo(upass)==0)
{
    msg.setText("Valid");
    System.out.println("Username is valid");
}
else
{
    throw new InvalidPasswordException();
}
catch(Exception e)
{
    msg.setText("Error");
}
attempt++;
}
else
{
    System.out.println("you are using 3 attempt");
    System.exit(0);
}
}
public static void main(String args[])
{
    PasswordDemo pd=new PasswordDemo();
    pd.login();
}
}
```

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.

SOLUTION:

```
import java.io.InputStreamReader;
import java.io.BufferedReader;
import java.io.IOException;
class Continent
{
String con;
InputStreamReader i = new InputStreamReader(System.in);
BufferedReader r = new BufferedReader(i);
void con_input() throws IOException
{
System.out.println("Enter Continent Name: ");
con = r.readLine();
}
}
class Country extends Continent
{
String cou ;
void cou_input() throws IOException
{
System.out.println("Enter Country Name: ");
cou = r.readLine();
}
}
class State extends Country
{
String sta;
void sta_input() throws IOException
{
System.out.println("Enter State Name: ");
sta = r.readLine();
}
}
class Main extends State
```

```

{
String pla;
void pla_input()throws IOException
{
System.out.println("Enter Place Name : ");
pla = r.readLine();

}

public static void main( String argsp[] )throws IOException
{
Main s = new Main();
s.con_input();
s.cou_input();
s.sta_input();
s.pla_input();
System.out.println("\n\nContinent: "+s.con);
System.out.println("Country: "+s.cou);
System.out.println("State: "+s.sta);
System.out.println("Place :"+ s.pla);
}
}

```

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

- Addition
- Multiplication
- Exit

SOLUTION:

```
package Practicals;
```

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

```
public class Ass1_setB2 {
```

```
    static BufferedReader sc = new BufferedReader(new
InputStreamReader(System.in));
```

```
// static int i,j;  
// return mat_add;
```

```
static int a[][] = new int[5][5];  
static int a2[][] = new int[5][5];
```

```
public static void main(String[] args) throws NumberFormatException,  
IOException {
```

```
    int m = 0, n = 0;
```

```
    System.out.println("-----Create two Matrix-----");
```

```
    System.out.print("Enter number of rows ");
```

```
    m = Integer.parseInt(sc.readLine());
```

```
    System.out.print("Enter number of columns ");
```

```
    n = Integer.parseInt(sc.readLine());
```

```
    if (m == n) {
```

```
        int i, j;
```

```
        System.out.println(
```

```
            "Enter values for 1st matrix: ");
```

```
        for (i = 0; i < m; i++) {
```

```
            for (j = 0; j < n; j++) {
```

```
                a[i][j] = Integer.parseInt(sc.readLine());
```

```
            }
```

```
        }
```

```
        System.out.println(
```

```
            "Enter values for 2nd matrix: ");
```

```
        for (i = 0; i < m; i++) {
```

```

        for (j = 0; j < n; j++) {
            a2[i][j] = Integer.parseInt(sc.readLine());
        }
    }

    System.out.println("1st Matrix :");
    for (i = 0; i < m; i++) {
        for (j = 0; j < n; j++) {
            System.out.print "[" + a[i][j] + " ";
        }
        System.out.println("");
    }

    System.out.println("2nd Matrix :");
    for (i = 0; i < m; i++) {
        for (j = 0; j < n; j++) {
            System.out.print "[" + a2[i][j] + " ";
        }
        System.out.println("");
    }

} else {
    System.out.println("Rows not equal to columns");
}

int ch;
do {
    System.out.println("\nChoose the matrix operation,");
    System.out.println("-----");

```

```
System.out.println("1. Addition");
System.out.println("2. Multiplication");
System.out.println("3. Transpose");
System.out.println("4. Exit");
System.out.println("-----");
System.out.print("Enter your choice: ");
ch = Integer.parseInt(sc.readLine());
switch (ch) {
    case 1:
        int mat_add[][] = new int[m][n];
        int i1, j1;
        System.out.println("Addition of Matrix 1 and Matrix 2 is ");
        for (i1 = 0; i1 < m; i1++) {
            for (j1 = 0; j1 < n; j1++) {
                mat_add[i1][j1] = a[i1][j1] + a2[i1][j1];
                System.out.print "[" + mat_add[i1][j1] + " ";
            }
            System.out.println("");
        }
        break;
    case 2:
        int product[][] = new int[m][n];
        System.out.println("Multiplication of Matrix 1 and Matrix 2 = ");
        int i2, j2;
        for (i2 = 0; i2 < m; i2++) {
```

```

    for (j2 = 0; j2 < n; j2++) {
        product[i2][j2] = 0;
        for (int k = 0; k < m; k++) {
            product[i2][j2] += a[i2][k] * a2[k][j2];
        }
        System.out.print "[" + product[i2][j2] + " ";
    }
    System.out.println("");
}
break;
case 3:
    int transpose[][] = new int[m][n];
    int transpose2[][] = new int[m][n];
    System.out.println("Transpose of Matrix 1 and 2 = ");
    int i3, j3;
    for (i3 = 0; i3 < m; i3++) {
        for (j3 = 0; j3 < n; j3++) {
            transpose[i3][j3] = a[j3][i3];
            transpose2[i3][j3] = a2[j3][i3];
            System.out.print "[" + transpose[i3][j3] + " ";
            System.out.print "[" + transpose2[i3][j3] + " ";
        }
        System.out.println("");
    }
    break;

```


case 4:

System.exit(0);

default:

System.out.println("Enter Valid choice !");

System.exit(0);

break;

}

} while (ch != 4);

}

}

SLIP 6

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

SOLUTION:

```
// package Practicals;
import java.util.*;

class Employee {
    int id;
    String name;
    String deptname;
    double salary;
    static int count;

    Employee(){
        id=0;
        name="Name";
        deptname="Comp sci";
        salary=20000;
    }

    Employee(int id , String name , String deptname , double salary){
        this.id=id;
        this.name=name;
        this.deptname=deptname;
        this.salary=salary;
        count++;
    }
}
```

```
}
```

```
void display() {
```

```
    System.out.println("[ID:" + id + " NAME: " + name + " DEPARTMENT NAME: " + deptname + " SALARY: " + salary + " ]");
```

```
}
```

```
}
```

```
public class Main {
```

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

```
        Scanner sc = new Scanner(System.in);
```

```
        System.out.println("How many Employees ");
```

```
        int number = sc.nextInt();
```

```
        Employee[] emp = new Employee[number];
```

```
        for(int j=0; j<number; j++)
```

```
        {
```

```
            System.out.println("Enter ID ");
```

```
            int id = sc.nextInt();
```

```
            System.out.println("Enter Name ");
```

```
            String name = sc.next();
```

```
            System.out.println("Enter Department name");
```

```
            String deptname = sc.next();
```

```
            System.out.println("Enter Salary");
```

```
            int salary = sc.nextInt();
```

```
emp[j] = new Employee(id,name,deptname,salary);
```

```
}
```

```
for(int j=0; j<number; j++)
```

```
{
```

```
    emp[j].display();
```

```
}
```

```
System.out.println("\nThere are total "+Employee.count+" Employees.");
```

```
}
```

```
}
```

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. Define methods accept and display in all cases. Create 3 objects each of Purchase Order and Sales Order and accept and display details.

SOLUTION:

```
import java.io.BufferedReader;
```

```
import java.io.IOException;
```

```
import java.io.InputStreamReader;
```

```
abstract class Order{
```

```
String id,description;
```

```
}
```

```
class PurchaseOrder extends Order{
```

```
String Customername,Vendorname;
```

```
public void accept() throws IOException{
    System.out.println("Enter the id,description,names of customers and vendors:");
    BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
    id=br.readLine();
    description=br.readLine();
    Customername=br.readLine();
    Vendorname=br.readLine();
}

public void display(){
    System.out.println("id: "+id);
    System.out.println("Description: "+description);
    System.out.println("Customername: "+Customername);
    System.out.println("Vendorname: "+Vendorname);
    System.out.println("-----");
}
}

class SalesOrder extends Order{
    String Customername,Vendorname;
    public void accept() throws IOException{
        System.out.println("Enter the id,description,names of customers and vendors:");
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        id=br.readLine();
        description=br.readLine();
        Customername=br.readLine();
```

```
Vendorname=br.readLine();
}
public void display(){
System.out.println("id: "+id);
System.out.println("Description: "+description);
System.out.println("Customername: "+Customername);
System.out.println("Vendorname: "+Vendorname);
System.out.println("-----");
}
}

public class Main {
public static void main(String [] args) throws IOException{

int i;
System.out.println("Select Any One: ");
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
System.out.println("1.Purchase Order");
System.out.println("2.Sales Order");
int ch=Integer.parseInt(br.readLine());
switch(ch){
case 1:
System.out.println("Enter the number of purchase Orders: ");
int n=Integer.parseInt(br.readLine());
PurchaseOrder [] l=new PurchaseOrder[n];
for(i=0;i<n;i++){
```

```
l[i]=new PurchaseOrder();
l[i].accept();
}
for(i=0;i<n;i++){
l[i].display();
System.out.println ("Object is created");
}
break;
case 2:
System.out.println("Enter the number of sales orders: ");
int m=Integer.parseInt(br.readLine());
SalesOrder [] h=new SalesOrder[m];
for(i=0;i<m;i++){
h[i]=new SalesOrder();
h[i].accept();
}
for(i=0;i<m;i++){
h[i].display();
System.out.println(" Object is created ");
}
break;

}

}

}
```

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

SOLUTION:

```
import java.util.Scanner;

class BankDetails {
    private String accno;
    private String name;
    private String acc_type;
    private long balance;
    Scanner sc = new Scanner(System.in);
    //method to open new account
    public void openAccount() {
        System.out.print("Enter Account No: ");
        accno = sc.next();
        System.out.print("Enter Account type: ");
        acc_type = sc.next();
        System.out.print("Enter Name: ");
        name = sc.next();
        System.out.print("Enter Balance: ");
        balance = sc.nextLong();
    }
    //method to display account details
```



```
public void showAccount() {  
    System.out.println("Name of account holder: " + name);  
    System.out.println("Account no.: " + accno);  
    System.out.println("Account type: " + acc_type);  
    System.out.println("Balance: " + balance);  
}  
  
//method to deposit money  
public void deposit() {  
    long amt;  
    System.out.println("Enter the amount you want to deposit: ");  
    amt = sc.nextLong();  
    balance = balance + amt;  
}  
  
//method to withdraw money  
public void withdrawal() {  
    long amt;  
    System.out.println("Enter the amount you want to withdraw: ");  
    amt = sc.nextLong();  
    if (balance >= amt) {  
        balance = balance - amt;  
        System.out.println("Balance after withdrawal: " + balance);  
    } else {  
        System.out.println("Your balance is less than " + amt + "\tTransaction  
failed...!!" );  
    }  
}
```

```
//method to search an account number
public boolean search(String ac_no) {
    if (accno.equals(ac_no)) {
        showAccount();
        return (true);
    }
    return (false);
}

public class BankingApp {
    public static void main(String arg[]) {
        Scanner sc = new Scanner(System.in);
        //create initial accounts
        System.out.print("How many number of customers do you want to input?
");
        int n = sc.nextInt();
        BankDetails C[] = new BankDetails[n];
        for (int i = 0; i < C.length; i++) {
            C[i] = new BankDetails();
            C[i].openAccount();
        }
        // loop runs until number 5 is not pressed to exit
        int ch;
        do {
            System.out.println("\n ***Banking System Application***");
```

```
System.out.println("1. Display all account details \n 2. Search by Account  
number\n 3. Deposit the amount \n 4. Withdraw the amount \n 5.Exit ");
```

```
System.out.println("Enter your choice: ");
```

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

```
switch (ch) {
```

```
    case 1:
```

```
        for (int i = 0; i < C.length; i++) {
```

```
            C[i].showAccount();
```

```
        }
```

```
        break;
```

```
    case 2:
```

```
        System.out.print("Enter account no. you want to search: ");
```

```
        String ac_no = sc.next();
```

```
        boolean found = false;
```

```
        for (int i = 0; i < C.length; i++) {
```

```
            found = C[i].search(ac_no);
```

```
            if (found) {
```

```
                break;
```

```
            }
```

```
        }
```

```
        if (!found) {
```

```
            System.out.println("Search failed! Account doesn't exist..!!");
```

```
        }
```

```
        break;
```

```
    case 3:
```

```
        System.out.print("Enter Account no. : ");
```

```
ac_no = sc.next();
found = false;
for (int i = 0; i < C.length; i++) {
    found = C[i].search(ac_no);
    if (found) {
        C[i].deposit();
        break;
    }
}
if (!found) {
    System.out.println("Search failed! Account doesn't exist..!!");
}
break;
case 4:
    System.out.print("Enter Account No : ");
    ac_no = sc.next();
    found = false;
    for (int i = 0; i < C.length; i++) {
        found = C[i].search(ac_no);
        if (found) {
            C[i].withdrawal();
            break;
        }
    }
}
if (!found) {
```

```
        System.out.println("Search failed! Account doesn't exist..!!");
    }
    break;
case 5:
    System.out.println("See you soon...");
    break;
}
}
while (ch != 5);
}
}
```

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.

SOLUTION:

THANGE KE PAS HAI

SLIP 8

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

SOLUTION:

```
import java.util.Scanner;
```

```
class Sphere {
```

```
    private double radius;
```

```
    public Sphere() {
```

```
        Scanner scanner = new Scanner(System.in);
```

```
        System.out.print("Enter the radius of the sphere: ");
```

```
        this.radius = scanner.nextDouble();
```

```
    }
```

```
    public double getRadius() {
```

```
        return radius;
```

```
    }
```

```
    public void setRadius(double radius) {
```

```
        this.radius = radius;
```

```
    }
```

```
    public double getVolume() {
```

```
        return (4.0 / 3.0) * Math.PI * Math.pow(radius, 3);
```

```

    }

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

public class Main{
    public static void main(String[] args){
        Sphere sphere = new Sphere();
        double volume = sphere.getVolume();
        double surfaceArea = sphere.getSurfaceArea();
        System.out.println("Volume: " + volume);
        System.out.println("Surface area: " + surfaceArea);
    }
}

```

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.

SOLUTION:

```

package Practicals;

import java.awt.*;
import java.awt.event.*;

class MyFrame extends Frame
{

```

```
TextField t,t1;
Label l,l1;
int x,y;
Panel p;
MyFrame(String title)
{
    super(title);
    setLayout(new FlowLayout());

    p=new Panel();
    p.setLayout(new GridLayout(2,2,5,5));
    t=new TextField(20);
    l= new Label("Co-ordinates of clicking");
    l1= new Label("Co-ordinates of movement");
    t1=new TextField(20);
    p.add(l);
    p.add(t);
    p.add(l1);
    p.add(t1);
    add(p);
    addMouseListener(new MyClick());
    addMouseMotionListener(new MyMove());
    setSize(500,500);
    setVisible(true);
}
```



```
class MyClick extends MouseAdapter
{
    public void mouseClicked(MouseEvent me)
    {
        x=me.getX();
        y=me.getY();
        t.setText("X="+x+" Y="+y);
    }
}

class MyMove extends MouseMotionAdapter
{
    public void mouseMoved(MouseEvent me)
    {
        x=me.getX();
        y=me.getY();
        t1.setText("X="+ x +" Y="+y);
    }
}

}

public class Ass5_SetA2
{
    public static void main(String args[])
    {
        new MyFrame("Mouse locator");
    } }
```

sLIP 9

Q1) 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

SOLUTION:

```
import java.util.Scanner;
class Clock {
    private int hours;
    private int minutes;
    private int seconds;
    private boolean isAM;
```

```
    public Clock(int hours, int minutes, int seconds) {
        this.hours = hours;
        this.minutes = minutes;
        this.seconds = seconds;
```

```
    // Copy code
    if (hours >= 12) {
        this.isAM = false;
    } else {
        this.isAM = true;
    }
}
```

```
    public void setTime(int hours, int minutes, int seconds) {
        if (hours >= 0 && hours <= 23 && minutes >= 0 && minutes <= 59 &&
            seconds >= 0 && seconds <= 59) {
            this.hours = hours;
            this.minutes = minutes;
            this.seconds = seconds;
```

```
    // Copy code
    if (hours >= 12) {
        this.isAM = false;
    } else {
```

```
        this.isAM = true;
    }
} else {
    System.out.println("Invalid time input. Please enter valid hours, minutes, and
seconds.");
}
}
```

```
public void displayTime() {
    System.out.print(hours + ":" + minutes + ":" + seconds + " ");
    if (isAM) {
        System.out.println("AM");
    } else {
        System.out.println("PM");
    }
}
}
```

```
public class Main{
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter hours: ");
        int hours = sc.nextInt();
        System.out.print("Enter minutes: ");
        int minutes = sc.nextInt();
        System.out.print("Enter seconds: ");
        int seconds = sc.nextInt();
```

```
// Copy code
        Clock clock = new Clock(hours, minutes, seconds);
        clock.displayTime();
```

```
        System.out.print("Enter new hours: ");
        hours = sc.nextInt();
        System.out.print("Enter new minutes: ");
        minutes = sc.nextInt();
        System.out.print("Enter new seconds: ");
        seconds = sc.nextInt();
```

```
        clock.setTime(hours, minutes, seconds);
        clock.displayTime();
```

```
}  
}
```

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 objects of class product and display the contents of each object and Also display the object count.

SOLUTION:

```
import java.util.Scanner;
```

```
class Product implements java.io.Serializable {
```

```
private int productId;
```

```
private String productName;
```

```
private double productCost;
```

```
private int productQuantity;
```

```
private static int objectCount = 0;
```

```
public Product() {
```

```
    this.productId = 0;
```

```
    this.productName = "";
```

```
    this.productCost = 0.0;
```

```
    this.productQuantity = 0;
```

```
    objectCount++;
```

```
}
```

```
public Product(int productId, String productName, double productCost, int  
productQuantity) {
```

```
    this.productId = productId;
```

```
    this.productName = productName;
```

```
this.productCost = productCost;  
this.productQuantity = productQuantity;  
objectCount++;  
}
```

```
public void displayProduct() {  
    System.out.println("=====");  
    System.out.println("Product ID: " + productId);  
    System.out.println("Product Name: " + productName);  
    System.out.println("Product Cost: " + productCost);  
    System.out.println("Product Quantity: " + productQuantity);  
    System.out.println("=====");  
}
```

```
public static void displayObjectCount() {  
    System.out.println("Number of Product objects: " + objectCount);  
}  
}
```

```
public class Main{  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        System.out.println("How many Products");  
        int number = sc.nextInt();  
        Product[] product1 = new Product[number];  
        for(int j=0; j<number; j++)  
        {
```

```
System.out.print("Enter product ID: ");
int productId = sc.nextInt();
sc.nextLine();
System.out.print("Enter product name: ");
String productName = sc.nextLine();
System.out.print("Enter product cost: ");
double productCost = sc.nextDouble();
System.out.print("Enter product quantity: ");
int productQuantity = sc.nextInt();
    product1[j] = new Product(productId, productName, productCost,
productQuantity);
}
for(int j=0; j<number; j++)
{
    product1[j].displayProduct();
}
Product product2 = new Product();
product2.displayProduct();

Product.displayObjectCount();

sc.close();
}
}
```

SLIP 10

Q1) Write a program to find the cube of given number using functional interface.

SOLUTION:

```
import java.util.Scanner;
```

```
@FunctionalInterface
interface Cube {
    int cube(int x);
}
```

```
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int num = sc.nextInt();
```

```
// Copy code
```

```
Cube cube = x -> x * x * x;
System.out.println("The cube of " + num + " is " + cube.cube(num));
```

```
sc.close();
}
}
```

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.

SOLUTION:

```
import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

class StudentInfo {

    int rollno;

    String name;

    String className;

    float percentage;

    StudentInfo(int rollno, String name, String className, float percentage) {

        this.rollno = rollno;

        this.name = name;

        this.className = className;

        this.percentage = percentage;

    }

    void display() {

        System.out.println("ROLL NO.: " + rollno);

        System.out.println("NAME: " + name);

        System.out.println("CLASS: " + className);

        System.out.println("PERCENTAGE: " + percentage);

    }

}

class StudentPer {
```



```
static float findPercentage(int[] marks) {  
    float total = 0;  
    for (int mark : marks) {  
        total += mark;  
    }  
    return total / marks.length;  
}  
  
public class Main {  
    public static void main(String[] args) throws IOException {  
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));  
        System.out.print("Enter roll number: ");  
        int rollno = Integer.parseInt(br.readLine());  
        System.out.print("Enter name: ");  
        String name = br.readLine();  
        System.out.print("Enter class: ");  
        String className = br.readLine();  
        int[] marks = new int[6];  
        for (int i = 0; i < 6; i++) {  
            System.out.print("Enter marks for subject " + (i+1) + ": ");  
            marks[i] = Integer.parseInt(br.readLine());  
        }  
        float percentage = StudentPer.findPercentage(marks);  
        StudentInfo student = new StudentInfo(rollno, name, className, percentage);  
        student.display();  
    }  
}
```

SLIP 11

Q1) 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.

SOLUTION:

```
import java.util.Scanner;
interface Operation {
    double volume();
}
```

```
class Cylinder implements Operation {
    public static final double PI = 3.142;
    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 Main {
    public static void main(String[] args) {
        // Read radius and height from user input
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter radius: ");
        double radius = scanner.nextDouble();
        System.out.print("Enter height: ");
        double height = scanner.nextDouble();
```

```
        // Create a new Cylinder object with user-specified radius and height
        Cylinder cylinder2 = new Cylinder(radius, height);
```

```
// Calculate and print the volume of the second cylinder
System.out.println("Volume of second cylinder: " + cylinder2.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.

SOLUTION:

```
import java.util.Scanner;

public class Main {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter username: ");
        String username = scanner.nextLine();

        System.out.print("Enter password: ");
        String password = scanner.nextLine();

        if (!username.equals(password)) {
            System.out.println("Invalid password");
        }

    }

}
```

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.

SOLUTION:

```
import java.util.Scanner;
class College {
    private int cno;
    private String cname;
    private String caddr;

    public College(int cno, String cname, String caddr) {
        this.cno = cno;
        this.cname = cname;
        this.caddr = caddr;
    }

    public void displayCollegeDetails() {
        System.out.println("College Number: " + cno);
        System.out.println("College Name: " + cname);
        System.out.println("College Address: " + caddr);
    }
}

class Department extends College {
    private int dno;
    private String dname;

    public Department(int cno, String cname, String caddr, int dno, String dname) {
        super(cno, cname, caddr);
        this.dno = dno;
        this.dname = dname;
    }

    @Override
    public void displayCollegeDetails() {
        super.displayCollegeDetails();
```

```
        System.out.println("Department Number: " + dno);
        System.out.println("Department Name: " + dname);
    }
}
```

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

        System.out.print("Enter college number: ");
        int cno = scanner.nextInt();

        System.out.print("Enter college name: ");
        String cname = scanner.next();

        System.out.print("Enter college address: ");
        String caddr = scanner.next();

        System.out.print("Enter department number: ");
        int dno = scanner.nextInt();

        System.out.print("Enter department name: ");
        String dname = scanner.next();

        Department department = new Department(cno, cname, caddr, dno, dname);
        department.displayCollegeDetails();
    }
}
```

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.

SOLUTION:

```
/* Program to create a Simple Calculator */
import java.awt.*;
import java.awt.event.*;
```

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

/*
<applet code="MyCalculator" width=300 height=300>
</applet>
*/

public class MyCalculator extends Applet implements ActionListener {
    int num1,num2,result;
    TextField T1;
    Button NumButtons[]=new Button[10];
    Button Add,Sub,Mul,Div,clear,EQ;
    char Operation;
    Panel nPanel,CPanel,SPanel;
    public void init() {
        nPanel=new Panel();
        T1=new TextField(30);
        nPanel.setLayout(new FlowLayout(FlowLayout.CENTER));
        nPanel.add(T1);
        CPanel=new Panel();
        CPanel.setBackground(Color.white);
        CPanel.setLayout(new GridLayout(5,5,3,3));
        for(int i=0;i<10;i++) {
            NumButtons[i]=new Button(""+i);
        }
        Add=new Button("+");
        Sub=new Button("-");
        Mul=new Button("*");
```

```
Div=new Button("/");
clear=new Button("clear");
EQ=new Button("=");
T1.addActionListener(this);
for(int i=0;i<10;i++) {
    CPanel.add(NumButtons[i]);
}
CPanel.add(Add);
CPanel.add(Sub);
CPanel.add(Mul);
CPanel.add(Div);
CPanel.add(EQ);
SPanel=new Panel();
SPanel.setLayout(new FlowLayout(FlowLayout.CENTER));
SPanel.setBackground(Color.yellow);
SPanel.add(clear);
for(int i=0;i<10;i++) {
    NumButtons[i].addActionListener(this);
}
Add.addActionListener(this);
Sub.addActionListener(this);
Mul.addActionListener(this);
Div.addActionListener(this);
clear.addActionListener(this);
EQ.addActionListener(this);
this.setLayout(new BorderLayout());
add(nPanel,BorderLayout.NORTH);
```

```
add(CPanel,BorderLayout.CENTER);
add(SPanel,BorderLayout.SOUTH);
}

public void actionPerformed(ActionEvent ae) {
    String str=ae.getActionCommand ();
    char ch=str.charAt(0);
    if(Character.isDigit(ch))
        T1.setText(T1.getText()+str);
    else
        if(str.equals("+")){
            num1=Integer.parseInt (T1.getText());
            Operation='+';
            T1.setText ("");
        }
        if(str.equals("-")){
            num1=Integer.parseInt(T1.getText());
            Operation='-';
            T1.setText("");
        }
        if(str.equals("*")){
            num1=Integer.parseInt(T1.getText());
            Operation='*';
            T1.setText("");
        }
        if(str.equals("/")){
            num1=Integer.parseInt(T1.getText());
            Operation='/';
```



```
T1.setText("");
}
if(str.equals("%")){
    num1=Integer.parseInt(T1.getText());
    Operation='%';
    T1.setText("");
}
if(str.equals("=")) {
    num2=Integer.parseInt(T1.getText());
    switch(Operation)
    {
        case '+':result=num1+num2;
            break;
        case '-':result=num1-num2;
            break;
        case '*':result=num1*num2;
            break;
        case '/':try {
            result=num1/num2;
        }
        catch(ArithmeticException e) {
            result=num2;
            JOptionPane.showMessageDialog(this,"Divided by zero");
        }
        break;
    }
}
T1.setText(""+result);
```

```

    }
    if(str.equals("clear")) {
        T1.setText("");
    }
}
}
}

```

SLIP 13

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

SOLUTION:

```

import java.io.BufferedReader;
import java.io.FileReader;

public class main
{
    public static void main(String[] args) throws Exception {
        String line;
        int count = 0;
        int lineCount = 0;

        //Opens a file in read mode
        FileReader file = new FileReader("data.txt");
        BufferedReader br = new BufferedReader(file);

        //Gets each line till end of file is reached
        while((line = br.readLine()) != null) {
            lineCount++;
            //Splits each line into words
            String words[] = line.split(" ");
            //Counts each word
            count = count + words.length;
        }

        System.out.println("Number of words present in given file: " + count);
    }
}

```

```

        System.out.println("Number of lines present in given file: " + lineCount);
        br.close();
    }
}

```

Q2) Write a program to display the system date and time in various formats shown below:

Current date is : 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

SOLUTION:

```

import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Locale;
public class NewClass {
    public static void main(String[] args) {
        Date date = new Date();
        SimpleDateFormat formatter = new SimpleDateFormat("dd/MM/yyyy");
        String strDate = formatter.format(date);
        System.out.println("Current date is: "+strDate);

        formatter = new SimpleDateFormat("MM-dd-yyyy");
        strDate = formatter.format(date);
        System.out.println("Current date is: "+strDate);

        formatter = new SimpleDateFormat("EEEEEE MMMM dd yyyy");
        strDate = formatter.format(date);
        System.out.println("Current date is: "+strDate);
    }
}

```

```
formatter = new SimpleDateFormat("E MMMM dd HH:mm:ss z yyyy");
strDate = formatter.format(date);
System.out.println("Current date and time is: "+strDate);
```

```
formatter = new SimpleDateFormat("dd/MM/yy HH:mm:ss a Z");
strDate = formatter.format(date);
System.out.println("Current date and time is: "+strDate);
```

```
formatter = new SimpleDateFormat("hh:mm:ss");
strDate = formatter.format(date);
System.out.println("Current time is: "+strDate);
```

```
formatter = new SimpleDateFormat("w");
strDate = formatter.format(date);
System.out.println("Current week of year is: "+strDate);
```

```
formatter = new SimpleDateFormat("W");
strDate = formatter.format(date);
System.out.println("Current week of the month is: "+strDate);
```

```
formatter = new SimpleDateFormat("D");
strDate = formatter.format(date);
System.out.println("Current day of the year: "+strDate);
```

```
}
}
```

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).

SOLUTION:

```
import java.util.Scanner;
```

```
public class Main {
```

```
    static class NumberIsZeroException extends Exception {
```

```
        public NumberIsZeroException(String message) {
```

```
            super(message);
```

```
        }
```

```
    }
```

```
    public static boolean isPrime(int number) {
```

```
        if (number < 2) {
```

```
            return false;
```

```
        }
```

```
        for (int i = 2; i <= Math.sqrt(number); i++) {
```

```
            if (number % i == 0) {
```

```
                return false;
```

```
            }
```

```
        }
```

```
        return true;
```

```
}
```

```
public static void main(String[] args) {  
    Scanner scanner = new Scanner(System.in);  
    System.out.print("Enter a number: ");  
    int number = scanner.nextInt();  
    try {  
        if (number == 0) {  
            throw new NumberIsZeroException("Number is 0");  
        }  
        if (isPrime(number)) {  
            System.out.println(number + " is a prime number.");  
        } else {  
            System.out.println(number + " is not a prime number.");  
        }  
    } catch (NumberIsZeroException e) {  
        System.out.println(e.getMessage());  
    }  
}
```

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.

SOLUTION:

Program 1:

```
package Assignment2.SY;

import java.io.BufferedReader;
import java.io.*;

public class SYClass {
    public int ct,mt,et;
    public void get() throws IOException{
        System.out.println("Enter marks of students for computer, maths and
        electronics subject out of 200 ");
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        ct=Integer.parseInt(br.readLine());
        mt=Integer.parseInt(br.readLine());
        et=Integer.parseInt(br.readLine());
    }

}
```

Program 2:

```
package Assignment2.TY;

import java.io.*;

public class TYClass {
```

```
public int tm,pm;

public void get() throws IOException{

    System.out.println("Enter the marks of the theory out of 400 and practicals out of 200: ");

    BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

    tm=Integer.parseInt(br.readLine());

    pm=Integer.parseInt(br.readLine());

}

}
```

Program 3:

```
package Assignment2;

import Assignment2.SY.*;
import Assignment2.TY.*;
import java.io.*;

class StudentInfo{

    int rollno;

    String name,grade;

    public float gt,tyt,syt;

    public float per;

    public void get() throws IOException{

        System.out.println("Enter roll number and name of the student: ");

        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

        rollno=Integer.parseInt(br.readLine());
```



```
name=br.readLine();  
}
```

```
}
```

```
public class StudentMarks {  
    public static void main(String[] args) throws IOException{  
  
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));  
        System.out.println("Enter the number of students:");  
        int n=Integer.parseInt(br.readLine());  
        SYClass sy[]=new SYClass[n];  
        TYClass ty[]=new TYClass[n];  
        StudentInfo si[]=new StudentInfo[n];  
        for(int i=0;i<n;i++)  
        {  
            si[i]=new StudentInfo();  
            sy[i]=new SYClass();  
            ty[i]=new TYClass();  
  
            si[i].get();  
            sy[i].get();  
            ty[i].get();  
  
            si[i].syt=sy[i].ct+sy[i].et+sy[i].mt;  
            si[i].tyt=ty[i].pm+ty[i].tm;
```

```
si[i].gt=si[i].syt+si[i].tyt;
```

```
si[i].per=(si[i].gt/1200)*100;
```

```
if(si[i].per>=70) si[i].grade="A";
```

```
else if(si[i].per>=60) si[i].grade="B";
```

```
else if(si[i].per>=50) si[i].grade="C";
```

```
else if(si[i].per>=40) si[i].grade="Pass";
```

```
else si[i].grade="Fail";
```

```
}
```

```
System.out.println("Roll  
No\tName\tSyTotal\tTyTotal\tGrandTotal\tPercentage\tGrade");
```

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

```
{
```

```
System.out.println(si[i].rollno+"\t"+si[i].name+"\t"+si[i].syt+"\t"+si[i].tyt+"\t"+si[i].gt+"\t"+si[i].per+"\t\t"+si[i].grade);
```

```
}
```

```
}
```

```
}
```

SLIP 15

Q1) 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.

SOLUTION:

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

```
public class FileCopier {  
    public static void main(String[] args) {  
        BufferedReader reader = new BufferedReader(new  
InputStreamReader(System.in));  
        try {  
            System.out.print("Enter the name of the first file: ");  
            String firstFileName = reader.readLine();  
            System.out.print("Enter the name of the second file: ");  
            String secondFileName = reader.readLine();  
  
            FileReader firstFileReader = new FileReader(firstFileName);  
            BufferedReader firstBufferedReader = new  
BufferedReader(firstFileReader);  
  
            FileWriter secondFileWriter = new FileWriter(secondFileName);  
            BufferedWriter secondBufferedWriter = new  
BufferedWriter(secondFileWriter);  
  
            String line;  
            while ((line = firstBufferedReader.readLine()) != null) {
```

```

        secondBufferedWriter.write(line);
        secondBufferedWriter.newLine();
    }

    firstBufferedReader.close();
    secondBufferedWriter.close();

    System.out.println("File copy completed successfully.");
} catch (IOException e) {
    System.out.println("An error occurred: " + e.getMessage());
}
}
}

```

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 withdrawamt. Write a appropriate method to display customer details.

SOLUTION:

```

class Account {
    String custname;
    int accno;

    // Default constructor
    Account() {
        custname = "";
        accno = 0;
    }
}

```

```
}
```

```
// Parameterized constructor
```

```
Account(String name, int number) {
```

```
    custname = name;
```

```
    accno = number;
```

```
}
```

```
}
```

```
class SavingAccount extends Account {
```

```
    double savingbal;
```

```
    double minbal;
```

```
// Default constructor
```

```
SavingAccount() {
```

```
    savingbal = 0.0;
```

```
    minbal = 0.0;
```

```
}
```

```
// Parameterized constructor
```

```
SavingAccount(String name, int number, double balance, double minimum) {
```

```
    super(name, number);
```

```
    savingbal = balance;
```

```
    minbal = minimum;
```

```
}
```

```
}
```

```
class AccountDetail extends SavingAccount {
```

```
    double depositamt;
```

```
    double withdrawamt;
```

```
// Default constructor
```

```
AccountDetail() {
```

```
    depositamt = 0.0;
```

```
    withdrawamt = 0.0;
```

```
}
```

```
// Parameterized constructor
```

```
AccountDetail(String name, int number, double balance, double minimum,  
double deposit, double withdrawal) {
```

```
    super(name, number, balance, minimum);
```

```
    depositamt = deposit;
```

```
    withdrawamt = withdrawal;
```

```
}
```

```
// Method to display customer details
```

```
void displayCustomerDetails() {
```

```
    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) {  
        // Create an object of the AccountDetail class  
        AccountDetail customer = new AccountDetail("John Doe", 123456, 1000.0,  
500.0, 200.0, 100.0);  
  
        // Call the displayCustomerDetails() method on the object  
        customer.displayCustomerDetails();  
    }  
}
```

SLIP 16

Q1) Write a program to find the Square of given number using function interface.

SOLUTION:

```
import java.util.Scanner;
```

```
@FunctionalInterface
```

```
interface Square {
```

```
    double getSquare(double num);
```

```
}
```

```
public class Main {
```

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

```
        Scanner scanner = new Scanner(System.in);
```

```
        // Read in a number from the user
```

```
        System.out.print("Enter a number: ");
```

```
        double num = scanner.nextDouble();
```

```
        // Create a lambda expression to calculate the square of the number
```

```
        Square square = (double x) -> x * x;
```

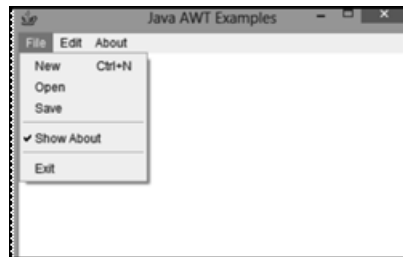
```
        // Call the getSquare() method on the lambda expression and print the result
```

```
        System.out.println("The square of " + num + " is " + square.getSquare(num));
```

```
}
```


}

Q2) Write a program to design a screen using Awt that,



SOLUTION:

SLIP 17

Q1) 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.

SOLUTION:

```
import java.util.Scanner;
```

```
class Customer {
```

```
    String name;
```

```
    String phoneNumber;
```

```
    // Default constructor
```

```
    Customer() {
```

```
        name = "";
```

```
        phoneNumber = "";
```

```
    }
```

```
    // Parameterized constructor
```

```
    Customer(String n, String pn) {
```

```
        name = n;
```

```
        phoneNumber = pn;
```

```
    }
```

```
}
```

```
class Depositor extends Customer {
```

```
int accno;  
double balance;
```

```
// Default constructor
```

```
Depositor() {  
    accno = 0;  
    balance = 0.0;  
}
```

```
// Parameterized constructor
```

```
Depositor(String n, String pn, int an, double b) {  
    super(n, pn);  
    accno = an;  
    balance = b;  
}  
}
```

```
class Borrower extends Depositor {
```

```
    int loanNo;  
    double loanAmt;
```

```
// Default constructor
```

```
Borrower() {  
    loanNo = 0;  
    loanAmt = 0.0;
```

```
}
```

```
// Parameterized constructor
```

```
Borrower(String n, String pn, int an, double b, int ln, double la) {
```

```
    super(n, pn, an, b);
```

```
    loanNo = ln;
```

```
    loanAmt = la;
```

```
}
```

```
}
```

```
public class Main {
```

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

```
        Scanner scanner = new Scanner(System.in);
```

```
        // Read in the number of customers
```

```
        System.out.print("Enter the number of customers: ");
```

```
        int n = scanner.nextInt();
```

```
        // Create an array of Borrower objects
```

```
        Borrower[] customers = new Borrower[n];
```

```
        // Read in the details of each customer
```

```
        for (int i = 0; i < n; i++) {
```

```
            System.out.println("Enter details for customer " + (i+1) + ":");
```

```
            System.out.print("Name: ");
```

```
String name = scanner.next();
System.out.print("Phone number: ");
String phoneNumber = scanner.next();
System.out.print("Account number: ");
int accno = scanner.nextInt();
System.out.print("Balance: ");
double balance = scanner.nextDouble();
System.out.print("Loan number: ");
int loanNo = scanner.nextInt();
System.out.print("Loan amount: ");
double loanAmt = scanner.nextDouble();

// Create a Borrower object for this customer and add it to the array
customers[i] = new Borrower(name, phoneNumber, accno, balance, loanNo,
loanAmt);
}
for (int i = 0; i < n; i++)
{
System.out.println("\nDetails for customer " + (i+1) + ":");
System.out.println("Name: " + customers[i].name);
System.out.println("Phone number: " + customers[i].phoneNumber);
System.out.println("Account number: " + customers[i].accno);
System.out.println("Balance: " + customers[i].balance);
System.out.println("Loan no.: " + customers[i].loanNo);
System.out.println("Loan Amount: " + customers[i].loanAmt);
}
```

```
}
```

```
}
```

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

SOLUTION:

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

class TextBoxesAndButtons extends JFrame {
    // Declare the text boxes and buttons
    private JTextField textField1;
    private JTextField textField2;
    private JTextField textField3;
    private JButton button1;
    private JButton button2;
    public TextBoxesAndButtons() {
        // Set the frame properties
        setTitle("Text Boxes and Buttons");
        setSize(400, 200);
        setLocationRelativeTo(null);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        // Create the text boxes and buttons
        textField1 = new JTextField();
        textField2 = new JTextField();
```

```
textField3 = new JTextField();  
button1 = new JButton("Concatenate");  
button2 = new JButton("Reverse");  
  
// Add action listeners to the buttons  
button1.addActionListener(new ConcatenateListener());  
button2.addActionListener(new ReverseListener());  
  
// Create a panel to hold the text boxes and buttons  
JPanel panel = new JPanel();  
panel.setLayout(new GridLayout(4, 2));  
  
// Add the text boxes and buttons to the panel  
panel.add(new JLabel("Enter a string:"));  
panel.add(textField1);  
panel.add(new JLabel("Enter another string:"));  
panel.add(textField2);  
panel.add(button1);  
panel.add(button2);  
panel.add(new JLabel("Result:"));  
panel.add(textField3);  
  
// Add the panel to the frame  
add(panel);  
}
```

// Action listener for the Concatenate button

class ConcatenateListener implements ActionListener {

public void actionPerformed(ActionEvent e) {

// Concatenate the strings and display the result in the third text box

String s1 = textField1.getText();

String s2 = textField2.getText();

textField3.setText(s1 + s2);

}

}

// Action listener for the Reverse button

class ReverseListener implements ActionListener {

public void actionPerformed(ActionEvent e) {

// Reverse the string and display the result in the third text box

String s = textField1.getText();

StringBuilder sb = new StringBuilder(s);

sb.reverse();

textField3.setText(sb.toString());

}

}

}

public class Main{

public static void main(String[] args) {

TextBoxesAndButtons frame = new TextBoxesAndButtons();


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

SLIP 18

Q1) Write a program to implement Border Layout Manager.

SOLUTION:

```
import java.awt.*;  
import javax.swing.*;  
  
public class Border  
{  
    JFrame f;  
    Border()  
{  
    f = new JFrame();  
  
    // creating buttons  
    JButton b1 = new JButton("NORTH");; // the button will be labeled as  
NORTH  
    JButton b2 = new JButton("SOUTH");; // the button will be labeled as  
SOUTH  
    JButton b3 = new JButton("EAST");; // the button will be labeled as EAST  
    JButton b4 = new JButton("WEST");; // the button will be labeled as  
WEST  
    JButton b5 = new JButton("CENTER");; // the button will be labeled as
```

CENTER

f.add(b1, BorderLayout.NORTH); // b1 will be placed in the North Direction

f.add(b2, BorderLayout.SOUTH); // b2 will be placed in the South Direction

f.add(b3, BorderLayout.EAST); // b2 will be placed in the East Direction

f.add(b4, BorderLayout.WEST); // b2 will be placed in the West Direction

f.add(b5, BorderLayout.CENTER); // b2 will be placed in the Center

f.setSize(300, 300);

f.setVisible(true);

}

public static void main(String[] args) {

new Border();

}

}

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.

SOLUTION:

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

class CricketPlayer {

```
String name;  
int no_of_innings;  
int no_of_times_not_out;  
int totlaruns;  
float bat_avg;  
static int count;
```

```
CricketPlayer() {  
}
```

```
CricketPlayer(String name, int no_of_innings, int no_of_times_not_out, int  
totlaruns , float bat_avg) {  
    count++;  
    this.name=name;  
    this.no_of_innings=no_of_innings;  
    this.no_of_times_not_out=no_of_times_not_out;  
    this.totlaruns=totlaruns;  
    this.bat_avg=bat_avg;  
}
```

```
void display() {  
    System.out.println("NAME: "+name+" INNINGS: "+no_of_innings+" NOT  
OUT: "+no_of_times_not_out+" TOTAL RUNS: "+totlaruns+" BATTING  
AVERAGE: "+bat_avg);  
}
```

```
float getBatAvg() {  
    return bat_avg;
```

```
}
```

```
void counter() {
```

```
    System.out.println(count + " object is created");
```

```
}
```

```
public static void sortStudent(CricketPlayer s[], int n) {
```

```
    for (int i = n - 1; i >= 0; i--) {
```

```
        for (int j = 0; j < i; j++) {
```

```
            if (s[j].getBatAvg() > s[j + 1].getBatAvg()) {
```

```
                CricketPlayer t = s[j];
```

```
                s[j] = s[j + 1];
```

```
                s[j + 1] = t;
```

```
            }
```

```
        }
```

```
    }
```

```
    System.out.println("-----List of Player sorted as per Batting Average-----");
```

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

```
        s[i].display();
```

```
}
```

```
}
```

```
public class ass2_SetB_q2 {
```

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

```
        BufferedReader br = new BufferedReader(new  
        InputStreamReader(System.in));
```

```
        System.out.println("Enter no. of Player:");
```

```
        int n = Integer.parseInt(br.readLine());
```

```

CricketPlayer p[] = new CricketPlayer[n];
for (int i = 0; i < n; i++) {
    System.out.print("Enter Name:");
    String name = br.readLine();
    System.out.print("Enter Innings:");
    int no_of_innings = Integer.parseInt(br.readLine());
    System.out.println("Enter number of times not out");
    int no_of_times_not_out = Integer.parseInt(br.readLine());
    System.out.println("Enter total runs");
    int totlaruns = Integer.parseInt(br.readLine());
    System.out.println("Enter Batting Average");
    float bat_avg = Float.parseFloat(br.readLine());

    p[i] = new
CricketPlayer(name,no_of_innings,no_of_times_not_out,totlaruns,bat_avg);
    p[i].counter();
}
CricketPlayer.sortStudent(p, CricketPlayer.count);
}
}

```

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

SOLUTION:

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

```
public class Main {
```

```
static BufferedReader sc = new BufferedReader(new  
InputStreamReader(System.in));
```

```
static int a[][] = new int[5][5];
```

```
public static void main(String[] args) throws NumberFormatException,  
IOException {
```

```
    int m = 0, n = 0;
```

```
    System.out.print("Enter number of rows ");
```

```
    m = Integer.parseInt(sc.readLine());
```

```
    System.out.print("Enter number of columns ");
```

```
    n = Integer.parseInt(sc.readLine());
```

```
    if (m == n) {
```

```
        int i, j;
```

```
        System.out.println("Enter values for matrix: ");
```

```
        for (i = 0; i < m; i++) {
```

```
            for (j = 0; j < n; j++) {
```

```
                a[i][j] = Integer.parseInt(sc.readLine());
```

```
            }
```

```
        }
```

```
        System.out.println("Original Matrix :");
```

```
        for (i = 0; i < m; i++) {
```

```
            for (j = 0; j < n; j++) {
```

```
                System.out.print "[" + a[i][j] + " " );
```

```
            }
```

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

```
    }
```

```
// Calculate and display the sum of the diagonal elements
int sum = 0;
for (i = 0; i < m; i++) {
    sum += a[i][i];
}
System.out.println("Sum of diagonal elements: " + sum);
} else {
    System.out.println("Rows not equal to columns");
}
}
}
```

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.

SOLUTION:

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.

SOLUTION:

```
import java.io.InputStreamReader;
import java.io.BufferedReader;
import java.io.IOException;
class Continent
{
String con;
InputStreamReader i = new InputStreamReader(System.in);
BufferedReader r = new BufferedReader(i);
void con_input() throws IOException
{
System.out.println("Enter Continent Name: ");
con = r.readLine();
}
}
class Country extends Continent
{
String cou ;
void cou_input() throws IOException
{
System.out.println("Enter Country Name: ");
cou = r.readLine();
}
}
class State extends Country
{
String sta;
void sta_input() throws IOException
{
System.out.println("Enter State Name: ");
sta = r.readLine();
}
}
```



```

class Main extends State
{
String pla;
void pla_input()throws IOException
{
System.out.println("Enter Place Name : ");
pla = r.readLine();

}

public static void main( String argsp[] )throws IOException
{
Main s = new Main();
s.con_input();
s.cou_input();
s.sta_input();
s.pla_input();
System.out.println("\n\nContinent: "+s.con);
System.out.println("Country: "+s.cou);
System.out.println("State: "+s.sta);
System.out.println("Place :"+ s.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 integers

SOLUTION:

```

package Operation;

```

```

import java.util.Scanner; //import Scanner class for user input

```

```

public class Addition {

//method to add two integers

```

```
public static int add(int num1, int num2) {  
    return num1 + num2;  
}
```

//method to subtract two float values

```
public static float subtract(float num1, float num2) {  
    return num1 - num2;  
}  
}
```

```
public class Maximum {
```

//method to find the maximum of two integers

```
public static int max(int num1, int num2) {  
    if (num1 > num2) {  
        return num1;  
    } else {  
        return num2;  
    }  
}  
}
```

```
public class TestOperation {
```

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

```
    Scanner scanner = new Scanner(System.in);
```

```
System.out.println("Enter two integers to add: ");  
int num1 = scanner.nextInt();  
int num2 = scanner.nextInt();  
int sum = Addition.add(num1, num2);  
System.out.println("The sum is: " + sum);
```

```
System.out.println("Enter two float values to subtract: ");  
float num3 = scanner.nextFloat();  
float num4 = scanner.nextFloat();  
float difference = Addition.subtract(num3, num4);  
System.out.println("The difference is: " + difference);
```

```
System.out.println("Enter two integers to find the maximum: ");  
int num5 = scanner.nextInt();  
int num6 = scanner.nextInt();  
int maximum = Maximum.max(num5, num6);  
System.out.println("The maximum is: " + maximum);
```

```
scanner.close(); //close the scanner to prevent resource leak
```

```
}
```

```
}
```

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.

SOLUTION:

```
import java.util.Scanner;
```

```
class MyDate {
```

```
int day, month, year;
```

```
// Copy code
```

```
public MyDate(int day, int month, int year) throws InvalidDateException {
```

```
    if (!isValidDate(day, month, year)) {
```

```
        throw new InvalidDateException("Invalid date!");
```

```
    }
```

```
    this.day = day;
```

```
    this.month = month;
```

```
    this.year = year;
```

```
}
```

```
private boolean isValidDate(int day, int month, int year) {
```

```
    // check if day is valid for the month
```

```
    if (day < 1 || day > 31) {
```

```
        return false;
```

```
    }
```

```
if (month == 2) {  
    // check if day is valid for February  
    if (day > 29) {  
        return false;  
    }  
    // check for leap year  
    if (day == 29) {  
        if (year % 4 != 0 || (year % 100 == 0 && year % 400 != 0)) {  
            return false;  
        }  
    }  
}  
  
if (month == 4 || month == 6 || month == 9 || month == 11) {  
    // check if day is valid for months with 30 days  
    if (day > 30) {  
        return false;  
    }  
}  
  
// check if month is valid  
if (month < 1 || month > 12) {  
    return false;  
}  
  
return true;  
}
```

```
public void displayDate() {  
    System.out.println(day + "/" + month + "/" + year);  
}  
}
```

```
public class Main{  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        System.out.println("Enter date (dd mm yyyy): ");  
        int day = sc.nextInt();  
        int month = sc.nextInt();  
        int year = sc.nextInt();  
        try {  
            MyDate date = new MyDate(day, month, year);  
            date.displayDate();  
        } catch (InvalidDateException e) {  
            System.out.println(e.getMessage());  
        }  
    }  
}
```

```
class InvalidDateException extends Exception {  
    public InvalidDateException(String message) {  
        super(message);  
    }  
}
```

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.

SOLUTION:

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

```
class Employee {
```

```
    int id;
```

```
    String name;
```

```
    String deptname;
```

```
    double salary;
```

```
    static int count;
```

```
    Employee(){
```

```
        id=0;
```

```
        name="Name";
```

```
        deptname="Comp sci";
```

```
        salary=20000;
```

```
    }
```

```
    Employee(int id , String name , String deptname , double salary){
```

```
        this.id=id;
```

```
        this.name=name;
```

```
        this.deptname=deptname;
```

```
        this.salary=salary;
```

```
        count++;
```

```
    }
```

```
void display() {  
    System.out.println("[ID:" + id + " NAME: " + name + " DEPARTMENT NAME: "  
+ deptname + " SALARY: " + salary + " ]");  
}
```

```
}
```

```
public class Main {  
    public static void main(String[] args) {  
  
        Scanner sc = new Scanner(System.in);  
        System.out.println("How many Employees ");  
        int number = sc.nextInt();  
        Employee[] emp = new Employee[number];  
        for(int j=0; j<number; j++)  
        {  
            System.out.println("Enter ID ");  
            int id = sc.nextInt();  
            System.out.println("Enter Name ");  
            String name = sc.next();  
            System.out.println("Enter Department name");  
            String deptname = sc.next();  
            System.out.println("Enter Salary");  
            int salary = sc.nextInt();
```



```
emp[j] = new Employee(id,name,deptname,salary);
```

```
}
```

```
for(int j=0; j<number; j++)
```

```
{
```

```
    emp[j].display();
```

```
}
```

```
System.out.println("\nThere are total "+Employee.count+" Employees.");
```

```
}
```

```
}
```

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).

SOLUTION:

```
import java.util.Scanner;
```

```
abstract class Shape {  
    int length, width;
```

```
// Copy code
```

```
public Shape(int length, int width) {  
    this.length = length;  
    this.width = width;  
}
```

```
abstract void printArea();  
}
```

```
class Rectangle extends Shape {  
    public Rectangle(int length, int width) {  
        super(length, width);  
    }
```

// Copy code

@Override

void printArea() {

int area = length * width;

System.out.println("The area of the rectangle is: " + area);

}

}

class Triangle extends Shape {

public Triangle(int length, int width) {

super(length, width);

}

// Copy code

@Override

void printArea() {

int area = (length * width) / 2;

System.out.println("The area of the triangle is: " + area);

}

}

class Circle extends Shape {

public Circle(int radius) {

super(radius, 0);

}

// Copy code

@Override

```
void printArea() {  
    double area = Math.PI * Math.pow(length, 2);  
    System.out.println("The area of the circle is: " + area);  
}  
}
```

```
public class Main {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        System.out.println("Enter shape (rectangle, triangle, circle): ");  
        String shape = sc.nextLine();  
        if (shape.equalsIgnoreCase("rectangle")) {  
            System.out.println("Enter length and width: ");  
            int length = sc.nextInt();  
            int width = sc.nextInt();  
            Rectangle rectangle = new Rectangle(length, width);  
            rectangle.printArea();  
        } else if (shape.equalsIgnoreCase("triangle")) {  
            System.out.println("Enter base and height: ");  
            int base = sc.nextInt();  
            int height = sc.nextInt();  
            Triangle triangle = new Triangle(base, height);
```

```
triangle.printArea();  
} else if (shape.equalsIgnoreCase("circle")) {  
    System.out.println("Enter radius: ");  
    int radius = sc.nextInt();  
    Circle circle = new Circle(radius);  
    circle.printArea();  
} else {  
    System.out.println("Invalid shape!");  
}  
}  
}
```

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 java.awt.Color;  
import java.awt.Graphics;  
import java.awt.event.MouseAdapter;  
import java.awt.event.MouseEvent;  
  
import javax.swing.JFrame;  
import javax.swing.JPanel;  
  
public class Main extends JPanel {  
    String event = "";
```

// Copy code

```
public Main() {  
    addMouseListener(new MouseHandler());  
    addMouseMotionListener(new MouseHandler());  
}
```

```
public void paintComponent(Graphics g) {  
    super.paintComponent(g);  
    g.setColor(Color.RED);  
    g.drawString(event, getWidth() / 2, getHeight() / 2);  
}
```

```
class MouseHandler extends MouseAdapter {  
    public void mouseClicked(MouseEvent e) {  
        event = "Mouse clicked";  
        repaint();  
    }  
}
```

```
public void mouseEntered(MouseEvent e) {  
    event = "Mouse entered";  
    repaint();  
}
```

```
public void mouseExited(MouseEvent e) {  
    event = "Mouse exited";  
}
```

```
    repaint();  
}
```

```
public void mousePressed(MouseEvent e) {  
    event = "Mouse pressed";  
    repaint();  
}
```

```
public void mouseReleased(MouseEvent e) {  
    event = "Mouse released";  
    repaint();  
}
```

```
public void mouseDragged(MouseEvent e) {  
    event = "Mouse dragged";  
    repaint();  
}
```

```
public void mouseMoved(MouseEvent e) {  
    event = "Mouse moved";  
    repaint();  
}  
}
```

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

```
JFrame frame = new JFrame("Mouse Events");  
frame.add(new Main());  
frame.setSize(300, 300);  
frame.setVisible(true);  
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
}  
}
```


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.

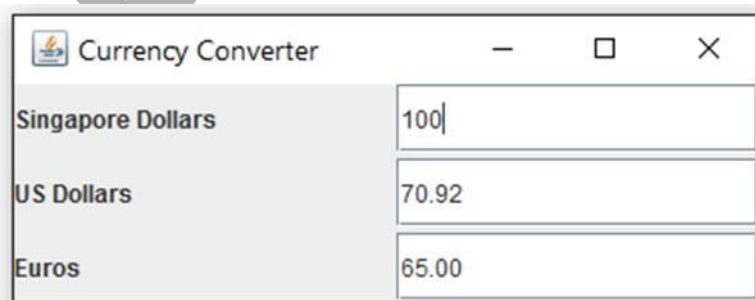
SOLUTION:

```
public class MyNumber {  
    private int x;  
    public MyNumber(){  
        x=0;  
    }  
    public MyNumber(int x){  
        this.x=x;  
    }  
    public boolean isNegative(){  
        if(x<0)  
            return true;  
        else return false;  
    }  
    public boolean isPositive(){  
        if(x>0)  
            return true;  
        else return false;  
    }  
    public boolean isZero(){  
        if(x==0)  
            return true;  
        else return false;  
    }  
    public boolean isOdd(){  
        if(x%2!=0)  
            return true;  
        else return false;  
    }  
    public boolean isEven(){  
        if(x%2==0)
```

```
    return true;
else return false;
}
```

```
public static void main(String [] args) throws
ArrayIndexOutOfBoundsException
{
    int x=Integer.parseInt(args[0]);
    MyNumber m=new MyNumber(x);
    if(m.isNegative())
        System.out.println("Number is Negative");
    if(m.isPositive())
        System.out.println("Number is Positive");
    if(m.isEven())
        System.out.println("Number is Even");
    if(m.isOdd())
        System.out.println("Number is Odd");
    if(m.isZero())
        System.out.println("Number is Zero");
}
}
```

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 = 0.65 Euro.



Currency Converter	
Singapore Dollars	100
US Dollars	70.92
Euros	65.00

SOLUTION:

```
import java.awt.EventQueue;
```

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

```
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextField;
```

```
public class Main extends JFrame {
    JTextField txtSGD, txtUSD, txtEuro;
    JButton btnConvert;
```

```
// Copy code
```

```
public Main() {
    JPanel panel = new JPanel();
    panel.setLayout(new GridLayout(4, 2));

    panel.add(new JLabel("Singapore Dollars: "));
    txtSGD = new JTextField();
    panel.add(txtSGD);

    panel.add(new JLabel("US Dollars: "));
    txtUSD = new JTextField();
```

```
panel.add(txtUSD);
```

```
panel.add(new JLabel("Euros: "));
```

```
txtEuro = new JTextField();
```

```
panel.add(txtEuro);
```

```
btnConvert = new JButton("Convert");
```

```
btnConvert.addActionListener(new ActionListener() {
```

```
    @Override
```

```
    public void actionPerformed(ActionEvent e) {
```

```
        if (!txtSGD.getText().isEmpty()) {
```

```
            double sgd = Double.parseDouble(txtSGD.getText());
```

```
            txtUSD.setText(String.format("%.2f", sgd / 1.41));
```

```
            txtEuro.setText(String.format("%.2f", sgd / 0.65));
```

```
        } else if (!txtUSD.getText().isEmpty()) {
```

```
            double usd = Double.parseDouble(txtUSD.getText());
```

```
            txtSGD.setText(String.format("%.2f", usd * 1.41));
```

```
            txtEuro.setText(String.format("%.2f", usd / 0.92));
```

```
        } else if (!txtEuro.getText().isEmpty()) {
```

```
            double euro = Double.parseDouble(txtEuro.getText());
```

```
            txtSGD.setText(String.format("%.2f", euro * 0.65));
```

```
            txtUSD.setText(String.format("%.2f", euro * 0.92));
```

```
        }
```

```
    }
```

```
});
```

```
panel.add(btnConvert);
```

```
add(panel);
```

```
setTitle("Currency Converter");
```

```
setSize(300, 150);
```

```
setLocationRelativeTo(null);
```

```
setDefaultCloseOperation(EXIT_ON_CLOSE);
```

```
}
```

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

```
    EventQueue.invokeLater(new Runnable() {
```

```
        @Override
```

```
        public void run() {
```

```
            Main ex = new Main();
```

```
            ex.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.

SOLUTION:

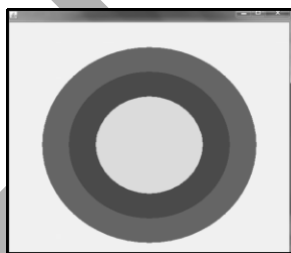
```
public class Main {  
    public static void main(String[] args) {  
        BankA bA = new BankA();  
        bA.getBalance();  
        BankB bB = new BankB();  
        bB.getBalance();  
        BankC bC = new BankC();  
        bC.getBalance();  
    }  
}  
  
abstract class Bank{  
    public abstract void getBalance();  
}  
  
class BankA extends Bank{  
    @Override  
    public void getBalance(){  
        System.out.println("Deposited: Rs.100 ");  
    }  
}
```

```
class BankB extends Bank{
    @Override
    public void getBalance(){
        System.out.println("Deposited: Rs.150 ");
    }
}
```

```
class BankC extends Bank{
    @Override
    public void getBalance(){
        System.out.println("Deposited: Rs.200 ");
    }
}
```

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.



SOLUTION:

```
import java.awt.*;
```

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

```
public class ConcentricCircles extends Frame implements MouseListener {
```

```
int x, y;
```

```
public ConcentricCircles() {  
    addMouseListener(this);  
    addWindowListener(new WindowAdapter() {  
        public void windowClosing(WindowEvent we) {  
            System.exit(0);  
        }  
    });  
}
```

```
public void paint(Graphics g) {  
    g.setColor(Color.BLACK);  
    g.drawOval(x-50, y-50, 100, 100);  
    g.drawOval(x-40, y-40, 80, 80);  
    g.drawOval(x-30, y-30, 60, 60);  
}
```

```
public void mouseClicked(MouseEvent me) {  
    x = me.getX();  
    y = me.getY();  
    repaint();  
}
```

```
public void mousePressed(MouseEvent me) {}
```



```
public void mouseReleased(MouseEvent me) {}  
public void mouseEntered(MouseEvent me) {}  
public void mouseExited(MouseEvent me) {}
```

```
public static void main(String[] args) {  
    ConcentricCircles cc = new ConcentricCircles();  
    cc.setSize(400, 400);  
    cc.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)

SOLUTION:

```
import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

class Student {

    int rollNo;

    String name;

    String studentClass;

    double percentage;

    public Student() throws IOException {

        BufferedReader reader = new BufferedReader(new

InputStreamReader(System.in));

        System.out.print("Enter roll number: ");

        rollNo = Integer.parseInt(reader.readLine());

        System.out.print("Enter name: ");

        name = reader.readLine();

        System.out.print("Enter class: ");
```

```
studentClass = reader.readLine();
```

```
System.out.print("Enter percentage: ");
```

```
percentage = Double.parseDouble(reader.readLine());
```

```
}
```

```
public void display() {
```

```
    System.out.println("Roll number: " + rollNo);
```

```
    System.out.println("Name: " + name);
```

```
    System.out.println("Class: " + studentClass);
```

```
    System.out.println("Percentage: " + percentage);
```

```
}
```

```
}
```

```
public class Main{
```

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

```
        Student s = new Student();
```

```
        s.display();
```

```
    }
```

```
}
```

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.

SOLUTION:

```
import javax.swing.*.*;
```

```
import java.awt.*.*;
```

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

```
class Swing2 extends JFrame implements ActionListener
```

```
{
```

```
    JLabel l1,l2,l3;
```

```
    JButton b;
```

```
    JRadioButton r1,r2,r3;
```

```
    JCheckBox c1,c2,c3;
```

```
    JTextField t1,t2;
```

```
    ButtonGroup b1;
```

```
    JPanel p1,p2;
```

```
    static int cnt;
```

```
    private StringBuffer s1=new StringBuffer();
```

```
    Swing2()
```

```
{
```

```
    b1=new ButtonGroup();
```

```
    p1=new JPanel();
```

```
    p2=new JPanel();
```

```
    b=new JButton("Clear");
```

```
    b.addActionListener(this);
```

```
    r1=new JRadioButton("FY");
```

```
    r2=new JRadioButton("SY");
```

```
    r3=new JRadioButton("TY");
```

```
b1.add(r1);  
b1.add(r2);  
b1.add(r3);  
r1.addActionListener(this);  
r2.addActionListener(this);  
r3.addActionListener(this);
```

```
c1=new JCheckBox("Music");  
c2=new JCheckBox("Dance");  
c3=new JCheckBox("Sports");
```

```
c1.addActionListener(this);  
c2.addActionListener(this);  
c3.addActionListener(this);
```

```
l1=new JLabel("Your Name");  
l2=new JLabel("Your Class");  
l3=new JLabel("Your Hobbies");  
t1=new JTextField(20);  
t2=new JTextField(30);
```

```
p1.setLayout(new GridLayout(5,2));  
p1.add(l1);p1.add(t1);  
p1.add(l2);p1.add(l3);  
p1.add(r1);p1.add(c1);  
p1.add(r2); p1.add(c2);  
p1.add(r3);p1.add(c3);
```

```
p2.setLayout(new FlowLayout());
```

```
p2.add(b);
```

```
p2.add(t2);
```

```
setLayout(new BorderLayout());
```

```
add(p1,BorderLayout.NORTH);
```

```
add(p2,BorderLayout.EAST);
```

```
setSize(400,200);
```

```
setVisible(true);
```

```
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
}
```

```
public void actionPerformed(ActionEvent e)
```

```
{
```

```
    if(e.getSource()==r1)
```

```
    {
```

```
        cnt++;
```

```
        if(cnt==1)
```

```
        {
```

```
            String s =t1.getText();
```

```
            s1.append("Name = ");
```

```
            s1.append(s);
```

```
        }
```

```
        s1.append(" Class = FY");
```

```
}  
else if(e.getSource()==r2)  
{  
    cnt++;  
    if(cnt==1)  
    {  
        String s =t1.getText();  
        s1.append("Name = ");  
        s1.append(s);  
    }  
    s1.append(" Class = SY");  
}  
else if(e.getSource()==r3)  
{  
    cnt++;  
    if(cnt==1)  
    {  
        String s =t1.getText();  
        s1.append("Name = ");  
        s1.append(s);  
    }  
    s1.append(" Class = TY");  
}
```

```
else if(e.getSource()==c1)  
{
```

```
s1.append(" Hobbies = Music");
}
else if(e.getSource()==c2)
{
    s1.append(" Hobbies = Dance");
}
else if(e.getSource()==c3)
{
    s1.append(" Hobbies = Sports");
}

t2.setText(new String(s1));
//      t2.setText(s2);

if(e.getSource()==b)
{
    t2.setText(" ");
    t1.setText(" ");
}
}
```

```
public static void main(String arg[])
{
    Swing2 s = new Swing2();

} }
```


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.

SOLUTION:

```
import java.util.Scanner;
```

```
class Item {
```

```
    // Static member to keep track of the number of objects created  
    private static int objectCount = 0;
```

```
    // Instance variables  
    private int itemNumber;  
    private String itemName;  
    private double itemPrice;
```

```
    // Default constructor  
    public Item() {  
        objectCount++;  
    }
```

```
    // Parameterized constructor  
    public Item(int itemNumber, String itemName, double itemPrice) {  
        this.itemNumber = itemNumber;
```

```
this.itemName = itemName;  
this.itemPrice = itemPrice;  
objectCount++;  
}
```

```
// Static method to get the object count  
public static int getObjectCount() {  
    return objectCount;  
}
```

```
// Method to display the contents of an Item object  
public void display() {  
    System.out.println("Item Number: " + itemNumber);  
    System.out.println("Item Name: " + itemName);  
    System.out.println("Item Price: " + itemPrice);  
}  
}
```

```
public class Main {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
  
        // Create an array of Item objects  
        Item[] items = new Item[3];
```

```
for (int i = 0; i < items.length; i++) {  
    // Get user input for the item number, name, and price  
    System.out.print("Enter item number: ");  
    int itemNumber = scanner.nextInt();  
    System.out.print("Enter item name: ");  
    String itemName = scanner.next();  
    System.out.print("Enter item price: ");  
    double itemPrice = scanner.nextDouble();  
  
    // Create a new Item object using the parameterized constructor  
    items[i] = new Item(itemNumber, itemName, itemPrice);  
  
    // Display the object count  
    System.out.println("Number of objects created: " +  
Item.getObjectCount());  
}  
  
// Display the contents of each Item object  
for (Item item : items) {  
    item.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 is 'A+ve' and had not donated for the recent six months.

Contact change and survive

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.

SOLUTION:

```
class Employee
{
    String name;
    int sal;
    Employee()
    {
        name=null;
        sal=0;
    }
    Employee(String n,int s)
    {
        name=n;
        sal=s;
    }
}
```

```
}  
int getSalary()  
{  
    return sal;  
}  
  
}
```

```
class Manager extends Employee
```

```
{  
    int hra,ta;
```

```
    Manager()
```

```
{  
    super();  
    hra=ta=0;  
}
```

```
    Manager(String n,int sal,int h,int t)
```

```
{  
    super(n,sal);  
    hra=h; ta=t;  
}
```

```
int getSalary()  
{
```

```
return (super.getSalary()+hra+ta);  
}  
}
```

```
class Main
```

```
{  
public static void main(String args[])  
{  
Manager m1=new Manager("Raj",20000,200,800); //parameter  
as(Name,salary,HRA,TA)  
System.out.println("Salary of Manager="+m1.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:

- i) If 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.
- ii) If it is a file display various details of that file.

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

```
class Main
```

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

```
String fname=a[0];
```

```
File f = new File(fname);
```

```
int num=0;
if(f.isDirectory())
{
    System.out.println("Given file "+fname+"is directory :");
    System.out.println("List of files are : ");
    String s[] = f.list();
    for(int i=0; i<s.length; i++)
    {
        File f1 = new File(fname, s[i]);

        if(f1.isFile())
        {
            num++;
            System.out.println(s[i]); //file name in directory
        }
        else System.out.println("\n"+s[i]+" is a sub directory");
    }
    System.out.println("\nNumber of files are: "+num);
}
else
{
    if(f.exists())
    {
        System.out.println("\n"+fname+" is a File");
        System.out.println("Details of "+fname+" are : ");
    }
}
```

```
        System.out.println("Path of file is "+f.getPath());
        System.out.println("Absolute Path of file is "+f.getAbsolutePath());
        System.out.println("Size of file is "+f.length());
    }
    else System.out.println(fname+" file is not present ");
}
}
```

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.

SOLUTION:

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

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the file name: ");
        String fileName = sc.nextLine();

        File file = new File(fileName);
```



```
// Check if the file exists
if (!file.exists()) {
    System.out.println("The file does not exist.");
} else {
    // Check if the file is readable
    if (file.canRead()) {
        System.out.println("The file is readable.");
    } else {
        System.out.println("The file is not readable.");
    }

    // Check if the file is writable
    if (file.canWrite()) {
        System.out.println("The file is writable.");
    } else {
        System.out.println("The file is not writable.");
    }

    // Check the type of file
    if (file.isDirectory()) {
        System.out.println("The file is a directory.");
    } else if (file.isFile()) {
        System.out.println("The file is a regular file.");
    } else {
        System.out.println("The file is not a regular file or a directory.");
    }
}
```

```

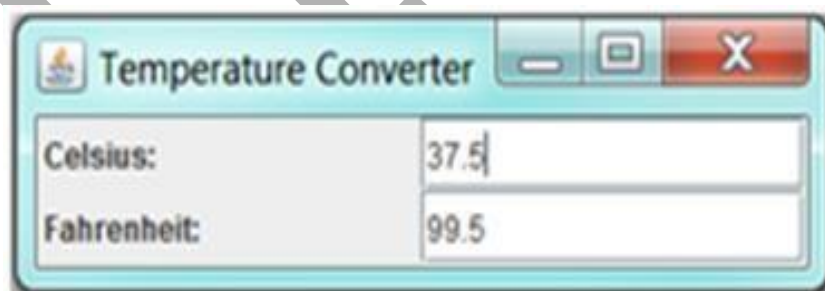
    }

    // Get the length of the file in bytes
    long length = file.length();

    System.out.println("The file is " + length + " bytes long.");
}
}
}

```

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"`.



SOLUTION:

File 1

```
import javax.swing.*;
```

```
import java.awt.*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.text.DecimalFormat;
```

```
public class CelsiusFahrenheitConverter extends JFrame implements  
ActionListener {
```

```
    private final JTextField jTextField;  
    private final JLabel result;
```

```
    public CelsiusFahrenheitConverter() {  
        super("Fahrenheit to Celsius converter");  
        setLayout(new FlowLayout());  
  
        jTextField = new JTextField("enter the value of Fahrenheit", 20);  
        add(jTextField);  
  
        JButton button = new JButton("OK");  
        button.addActionListener(this);  
        add(button);  
  
        result = new JLabel("Enter Fahrenheit value, and click OK");  
        add(result);  
    }
```

```
    public void actionPerformed(ActionEvent e) {
```

```

try {
    DecimalFormat decimalFormat = new DecimalFormat("#.##");
    double inputFahrenheit = Double.parseDouble(jTextField.getText());
    double answer = 0.0;

    answer = ((5.0 / 9.0) * (inputFahrenheit - 32.0));

    result.setText(String.valueOf(decimalFormat.format(answer)));
} catch (NumberFormatException ex) {
    System.out.println("You have entered not a number, but illegal symbols
or words, try again");
}
}
}

```

File 2

```

// CelsiusFahrenheitConverterRunner.java
// package FahrenheitTask;

```

```

import javax.swing.*;

```

```

public class CelsiusFahrenheitConverterRunner {
    public static void main(String[] args) {
        CelsiusFahrenheitConverter celsiusFahrenheitConverter = new
        CelsiusFahrenheitConverter();
    }
}

```

```
celsiusFahrenheitConverter.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
celsiusFahrenheitConverter.setSize(400, 250);  
celsiusFahrenheitConverter.setVisible(true);  
celsiusFahrenheitConverter.setLocationRelativeTo(null);  
}  
}
```

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.

SOLUTION:

```
import java.util.Scanner;
```

```
class Customer {
```

```
    public int custno;
```

```
    public String custname;
```

```
    public String contactnumber;
```

```
    public String custaddr;
```

```
    public Customer(int custno, String custname, String contactnumber, String custaddr) {
```

```
        this.custno = custno;
```

```
        this.custname = custname;
```

```
        this.contactnumber = contactnumber;
```

```
        this.custaddr = custaddr;
```

```
    }
```

```
    public static Customer searchByContactNumber(Customer[] customers, String contactNumber) {
```

```
        for (Customer customer : customers) {
```

```
            if (customer.contactnumber.equals(contactNumber)) {
```

```
                return customer;
```

```
    }  
}  
return null;  
}  
}
```

```
public class Main {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        System.out.print("Enter the number of customers: ");  
        int n = scanner.nextInt();  
  
        Customer[] customers = new Customer[n];  
        for (int i = 0; i < n; i++) {  
            System.out.println("Enter details for customer " + (i+1) + ":");  
            System.out.print("Customer number: ");  
            int custno = scanner.nextInt();  
            scanner.nextLine(); // to consume the remaining newline character  
            System.out.print("Customer name: ");  
            String custname = scanner.nextLine();  
            System.out.print("Contact number: ");  
            String contactnumber = scanner.nextLine();  
            System.out.print("Customer address: ");  
            String custaddr = scanner.nextLine();  
        }  
    }  
}
```

```
        customers[i] = new Customer(custno, custname, contactnumber,
custaddr);
    }
```

```
    System.out.print("Enter the contact number to search for: ");
    String contactNumber = scanner.nextLine();
    Customer customer = Customer.searchByContactNumber(customers,
contactNumber);
    if (customer == null) {
        System.out.println("No customer found with the given contact
number.");
    } else {
        System.out.println("Customer details:");
        System.out.println("Customer number: " + customer.custno);
        System.out.println("Customer name: " + customer.custname);
        System.out.println("Contact number: " + customer.contactnumber);
        System.out.println("Customer address: " + customer.custaddr);
    }
}
}
```

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.

SOLUTION:


```
import java.util.Scanner;
```

```
class Vehicle {  
    String company;  
    int price;  
}
```

```
class LightMotorVehicle extends Vehicle {  
    int mileage;  
}
```

```
class HeavyMotorVehicle extends Vehicle {  
    int capacityInTons;  
}
```

```
public class Main {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        System.out.print("Enter number of vehicles: ");  
        int n = scanner.nextInt();
```

```
        Vehicle[] vehicles = new Vehicle[n];
```

```
        for (int i = 0; i < n; i++) {
```

```
            System.out.print("Enter type of vehicle (1 for Light Motor Vehicle, 2 for  
Heavy Motor Vehicle): ");
```

```
int type = scanner.nextInt();

if (type == 1) {
    vehicles[i] = new LightMotorVehicle();
    System.out.print("Enter company name: ");
    vehicles[i].company = scanner.next();
    System.out.print("Enter price: ");
    vehicles[i].price = scanner.nextInt();
    System.out.print("Enter mileage: ");
    ((LightMotorVehicle)vehicles[i]).mileage = scanner.nextInt();
} else if (type == 2) {
    vehicles[i] = new HeavyMotorVehicle();
    System.out.print("Enter company name: ");
    vehicles[i].company = scanner.next();
    System.out.print("Enter price: ");
    vehicles[i].price = scanner.nextInt();
    System.out.print("Enter capacity in tons: ");
    ((HeavyMotorVehicle)vehicles[i]).capacityInTons = scanner.nextInt();
}
}
```

```
System.out.println("\nVehicle Information: ");
for (int i = 0; i < n; i++) {
    if (vehicles[i] instanceof LightMotorVehicle) {
        System.out.println("Type: Light Motor Vehicle");
    }
}
```

```
        System.out.println("Company: " + vehicles[i].company);  
        System.out.println("Price: " + vehicles[i].price);  
        System.out.println("Mileage: " +  
((LightMotorVehicle)vehicles[i]).mileage);  
    } else if (vehicles[i] instanceof HeavyMotorVehicle) {  
        System.out.println("Type: Heavy Motor Vehicle");  
        System.out.println("Company: " + vehicles[i].company);  
        System.out.println("Price: " + vehicles[i].price);  
        System.out.println("Capacity in tons: " +  
((HeavyMotorVehicle)vehicles[i]).capacityInTons);  
    }  
}  
}
```

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).

SOLUTION:

```
import java.util.Scanner;
```

```
class Person {
```

```
    String personName;
```

```
    String aadharNo;
```

```
    String panNo;
```

```
    Person(String personName, String aadharNo, String panNo) {
```

```
        this.personName = personName;
```

```
        this.aadharNo = aadharNo;
```

```
        this.panNo = panNo;
```

```
    }
```

```
    void displayInformation() {
```

```
        System.out.println("Person name: " + this.personName);
```

```
        System.out.println("Aadhar number: " + this.aadharNo);
```

```
        System.out.println("PAN number: " + this.panNo);
```

```
    }
```

```
}
```

```
class Main {  
    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.println("Enter information for person " + (i + 1));  
            System.out.print("Enter person name: ");  
            String personName = scanner.nextLine();  
            System.out.print("Enter Aadhar number: ");  
            String aadharNo = scanner.nextLine();  
            System.out.print("Enter PAN number: ");  
            String panNo = scanner.nextLine();  
  
            persons[i] = new Person(personName, aadharNo, panNo);  
        }  
  
        for (int i = 0; i < 5; i++) {  
            System.out.println("\nInformation for person " + (i + 1));  
            persons[i].displayInformation();  
        }  
    }  
}
```

Q2) Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Number1 and Number2. The division of Number1 and Number2 is displayed in the Result field when the Divide button is clicked. If Number1 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.

SOLUTION:

BS me gaya