**Slip 1**

**Q1) Write a Program to print all Prime numbers in an array of ‘n’ elements.**

**(use command line arguments) [10 marks]**

import java.util.Scanner;  
public class Prime Numbers{  
public static void main (String[] args){  
int[] array = new int [5];  
Scanner in = new Scanner (System.in);  
  
System.out.println("Enter the elements of the array: ");  
for(int i=0; i<5; i++)  
{  
array[i] = in.nextInt();  
}

for(int i=0; i<array.length; i++){  
boolean isPrime = true;

for (int j=2; j<i; j++){  
  
if(i%j==0){  
isPrime = false;  
break;  
}  
}

if(isPrime)  
  
System.out.println(i + " are the prime numbers in the array ");  
}  
}  
}

**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.**

import java.io.IOException;

import java.io.InputStreamReader;

abstract class Staff{

String name,address;

}

class FullTimeStaff extends Staff{

String department;

double salary;

public void accept() throws IOException{

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

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

name=br.readLine();

address=br.readLine();

department=br.readLine();

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

}

public void display(){

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

System.out.println("Address: "+address);

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, address, No of working hours and rate per hour: ");

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

name=br.readLine();

address=br.readLine();

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

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

}

public void display(){

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

System.out.println("Address: "+address);

System.out.println("No of Working Hours: "+hours);

System.out.println("Rate per hour: "+rate);

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

}

}

public class sb1 {

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 Satff");

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. In kgs / (ht)2)

import java.util.Scanner;

public class Example {

   public static void main(String args[]) {

      Scanner sc = new Scanner(System.in);

      System.out.print("Input weight in kilogram: ");

      double weight = sc.nextDouble();

      System.out.print("  
Input height in meters: ");

      double height = sc.nextDouble();

      double BMI = weight / (height \* height);

      System.out.print("  
The Body Mass Index (BMI) is " + BMI + " kg/m2");

   }

}

**Q2) Define a class CricketPlayer (name,no\_of\_innings,no\_of\_times\_notout, totatruns, bat\_avg). Create an array of n player objects .Calculate the batting average for each player using static method avg(). Define a static sort method which sorts the array on the basis of average. Display the player details in sorted order.**

import java.io.\*;

class Cricket {

String name;

int inning, tofnotout, totalruns;

float batavg;

public Cricket(){

name=null;

inning=0;

tofnotout=0;

totalruns=0;

batavg=0;

}

public void get() throws IOException{

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

System.out.println("Enter the name, no of innings, no of times not out, total runs: ");

name=br.readLine();

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

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

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

}

public void put(){

System.out.println("Name="+name);

System.out.println("no of innings="+inning);

System.out.println("no times notout="+tofnotout);

System.out.println("total runs="+totalruns);

System.out.println("bat avg="+batavg);

}

static void avg(int n, Cricket c[]){

try{

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

c[i].batavg=c[i].totalruns/c[i].inning;

}

}catch(ArithmeticException e){

System.out.println("Invalid arg");

}

}

static void sort(int n, Cricket c[]){

String temp1;

int temp2,temp3,temp4;

float temp5;

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

for(int j=i+1;j<n;j++){

if(c[i].batavg<c[j].batavg){

temp1=c[i].name;

c[i].name=c[j].name;

c[j].name=temp1;

temp2=c[i].inning;

c[i].inning=c[j].inning;

c[j].inning=temp2;

temp3=c[i].tofnotout;

c[i].tofnotout=c[j].tofnotout;

c[j].tofnotout=temp3;

temp4=c[i].totalruns;

c[i].totalruns=c[j].totalruns;

c[j].totalruns=temp4;

temp5=c[i].batavg;

c[i].batavg=c[j].batavg;

c[j].batavg=temp5;

}

}

}

}

}

public class a4sa1 {

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

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

System.out.println("Enter the limit:");

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

Cricket c[]=new Cricket[n];

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

c[i]=new Cricket();

c[i].get();

}

Cricket.avg(n,c);

Cricket.sort(n, c);

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

c[i].put();

}

}

}

**Slip 3**

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

//Java program to count words in a string.

import java.util.Scanner;

class SortStrings

{

public static void main(String args[])

{

String temp;

Scanner SC = new Scanner(System.in);

System.out.print("Enter the value of N: ");

int N= SC.nextInt();

SC.nextLine(); //ignore next line character

String names[] = new String[N];

System.out.println("Enter names: ");

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

{

System.out.print("Enter name [ " + (i+1) +" ]: ");

names[i] = SC.nextLine();

}

//sorting strings

for(int i=0; i<5; i++)

{

for(int j=1; j<5; j++)

{

if(names[j-1].compareTo(names[j])>0)

{

temp=names[j-1];

names[j-1]=names[j];

names[j]=temp;

}

}

}

System.out.println("\nSorted names are in Ascending Order: ");

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

{

System.out.println(names[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**.

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.**

import java.util.Scanner;

public class Solution {

public static void main(String[] args) {int[][] twodm = {{10, 20, 30}

{40, 50, 60}};

System.out.print("Original Array:\n");

print\_array(twodm);

System.out.print("After changing the rows and columns of the said array:");

transpose(twodm);

}

private static void transpose(int[][] twodm) {

int[][] newtwodm = new int[twodm[0].length][twodm.length];

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

for (int j = 0; j < twodm[0].length; j++) {

newtwodm[j][i] = twodm[i][j];

}

}

print\_array(newtwodm);

}

private static void print\_array(int[][] twodm) {

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

for (int j = 0; j < twodm[0].length; j++) {

System.out.print(twodm[i][j] + " ");

}

System.out.println();

}

}

}

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

.

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

public class facebookq extends JFrame implements ActionListener{

JPanel p,p1,p2;

JLabel name,pass;

String a,b;

int attempts=0;

JTextField t1,t2;

JButton b1,b2;

    public facebookq() {

    setTitle("Facebook");

    setSize(800,760);

    GridBagLayout gb=new GridBagLayout();

    GridBagConstraints c = new GridBagConstraints();

    setLayout(gb);

    window close=new window();

    addWindowListener(close);

    p = new JPanel();

    c.gridx=1; c.gridy=1;

    gb.setConstraints(p,c);

    add(p);

    name = new JLabel("username");

    c.gridx=1; c.gridy=1;

    gb.setConstraints(name,c);

    p.add(name);

    t1 = new JTextField(20);

    c.gridx=1; c.gridy=2;

    gb.setConstraints(t1,c);

    p.add(t1);

    p1 = new JPanel();

    c.gridx=1; c.gridy=2;

    gb.setConstraints(p1,c);

    add(p1);

    pass= new JLabel("Password");

    c.gridx=1; c.gridy=2;

    gb.setConstraints(pass,c);

    p1.add(pass);

    t2 = new JTextField(20);

    c.gridx=1; c.gridy=2;

    gb.setConstraints(t2,c);

    p1.add(t2);

    p2 = new JPanel();

    c.gridx=1; c.gridy=3;

    gb.setConstraints(p2,c);

    add(p2);

    b1=new JButton("LOGIN");

    c.gridx=1; c.gridy=3;

    gb.setConstraints(b1,c);

    p2.add(b1);

    b2=new JButton("CLEAR");

    c.gridx=1; c.gridy=4;

    gb.setConstraints(b2,c);

    p2.add(b2);

    b1.addActionListener(this);

    b2.addActionListener(this);

}

class LoginException extends Exception {

   LoginException() {

JOptionPane.showMessageDialog(null, "Exception!!! Invalid Username or Password");

   }

}

public void actionPerformed(ActionEvent e)

{                                    //open main

String cmd=e.getActionCommand();

try{                         //open try

if (cmd.equals("LOGIN"))

{

a=t1.getText();

b=t2.getText();

if ((a.equals("mk")) && (b.equals("mk")))

{

JOptionPane.showMessageDialog(null, "Suffesfully Done");

}

else

 {

 attempts++;

 if(attempts>=3)

{

 b1.setEnabled(false);

JOptionPane.showMessageDialog(null, "Limit is exceeded Try After Sometime");

}

else

{

throw new LoginException();

}

}

}

} //close try

catch (Exception er){}

if (cmd.equals("CLEAR"))

{

t1.setText("");

t2.setText("");

}

} //close main

public class window extends WindowAdapter

{

public void windowClosing(WindowEvent e)

{

    System.exit(0);

}

}

public static void main(String a[])

{

facebookq fb=new facebookq();

fb.setVisible(true);

}

}

**Slip 5**

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

import java.io.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

code:-

import java.util.Scanner;

class ArrayOperations {

public static void main(String[] args) {

int ch = -1;

int row1,row2,col1,col2;

Scanner s = new Scanner(System.in);

do {

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

System.out.println("1.MATRIX ADDITION");

System.out.println("2.MATRIX MULTIPLICATION");

System.out.println("0.EXIT");

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

System.out.print("Enter Your Choice : ");

ch = s.nextInt();

switch (ch) {

case 1:

System.out.print("Enter how many rows you want Matrix-1 : ");

row1 = s.nextInt();

System.out.print("Enter how many rows you want Matrix-2 : ");

row2 = s.nextInt();

System.out.print("Enter how many cols you want Matrix-1 : ");

col1 = s.nextInt();

System.out.print("Enter how many cols you want Matrix-2 : ");

col2 = s.nextInt();

if (row1 != row2 || col1 != col2) {

System.out.println("Matrix Addition is not possible");

} else {

int mat1[][] = new int[row1][col1];

int mat2[][] = new int[row2][col2];

int add[][] = new int[row1][col1];

for (int i = 0; i < row1; i++) {

for (int j = 0; j < col1; j++) {

System.out.print("Enter Matrix-1[" + i + "," + j + "] : ");

mat1[i][j] = s.nextInt();

}

}

for (int i = 0; i < row2; i++) {

for (int j = 0; j < col2; j++) {

System.out.print("Enter Matrix-2[" + i + "," + j + "] : ");

mat2[i][j] = s.nextInt();

}

}

for (int i = 0; i < row2; i++) {

for (int j = 0; j < col2; j++) {

add[i][j] = mat1[i][j] + mat2[i][j];

}

}

System.out.println();

for (int i = 0; i < row1; i++) {

for (int j = 0; j < col1; j++) {

System.out.print(mat1[i][j] + " ");

}

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

for (int j = 0; j < col1; j++) {

System.out.print(mat2[i][j] + " ");

}

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

for (int j = 0; j < col1; j++) {

System.out.print(add[i][j] + " ");

}

System.out.println();

}

}

break;

case 2:

System.out.print("Enter how many rows you want Matrix-1 : ");

row1 = s.nextInt();

System.out.print("Enter how many cols you want Matrix-1 : ");

col1 = s.nextInt();

System.out.print("Enter how many rows you want Matrix-2 : ");

row2 = s.nextInt();

System.out.print("Enter how many cols you want Matrix-2 : ");

col2 = s.nextInt();

if (col1 != row2) {

System.out.println("Matrix Multiplication is not possible");

} else {

int mat1[][] = new int[row1][col1];

int mat2[][] = new int[row2][col2];

int mul[][] = new int[row1][col2];

for (int i = 0; i < row1; i++) {

for (int j = 0; j < col1; j++) {

System.out.print("Enter Matrix-1[" + i + "," + j + "] : ");

mat1[i][j] = s.nextInt();

}

}

for (int i = 0; i < row2; i++) {

for (int j = 0; j < col2; j++) {

System.out.print("Enter Matrix-2[" + i + "," + j + "] : ");

mat2[i][j] = s.nextInt();

}

}

for (int i = 0; i < row1; i++) {

for (int j = 0; j < col2; j++) {

for (int k = 0; k < col1; k++) {

mul[i][j] += mat1[i][k] \* mat2[k][j];

}

}

}

System.out.println();

for (int i = 0; i < row1; i++) {

for (int j = 0; j < col1; j++) {

System.out.print(mat1[i][j] + " ");

}

System.out.println();

}

System.out.println();

for (int i = 0; i < row2; i++) {

for (int j = 0; j < col2; j++) {

System.out.print(mat2[i][j] + " ");

}

System.out.println();

}

System.out.println();

for (int i = 0; i < row1; i++) {

for (int j = 0; j < col2; j++) {

System.out.print(mul[i][j] + " ");

}

System.out.println();

}

System.out.println();

}

break;

case 0:

break;

}

} while (ch != 0);

}

}

**Slip 6**

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

**package** JavaTpoint.JavaObjectToJSON;

//Creating Employee class

**class** EmployeeDetails {

    //Creating properties of Employee class

**int** emp\_id, salary;

String name, address, department, email;

//Getter and setters for getting and setting properties

**public** **int** getEmp\_id() {

**return** emp\_id;

}

**public** **void** setEmp\_id(**int** emp\_id) {

**this**.emp\_id = emp\_id;

}

**public** **int** getSalary() {

**return** salary;

}

**public** **void** setSalary(**int** salary) {

**this**.salary = salary;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** String getAddress() {

**return** address;

}

**public** **void** setAddress(String address) {

**this**.address = address;

}

**public** String getDepartment() {

**return** department;

}

**public** **void** setDepartment(String department) {

**this**.department = department;

}

**public** String getEmail() {

**return** email;

}

**public** **void** setEmail(String email) {

**this**.email = email;

}

//Overriding toString() method

@Override

**public** String toString() {

**return** "Employee [emp\_id = " + emp\_id + ", salary = " + salary + ", name = " + name + ", address = " + address

            + ", department = " + department + ", email = " + email + "]";

    }

}

//Creating main class

**public** **class** Employee{

    //main() method start

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

        //Creating object of EmployeeDetails class

        EmployeeDetails emp = **new** EmployeeDetails();

        //Setting values to the properties

        emp.setEmp\_id(101);

        emp.setName("Emma Watson");

        emp.setDepartment("IT");

        emp.setSalary(15000);

        emp.setAddress("New Delhi");

        emp.setEmail("Emmawatson123@gmail.com");

        //Showing Employee details

        System.out.println(emp);

        //Getting salary using getter

**int** sal = emp.getSalary();

**int** increment = 0;

        //Incrementing salary based on condition

**if** ((sal >=1000) && (sal <=1500))

        {

            //incrementing salary 2%

            increment += (sal \* 2)/100;

            sal = sal+increment;

            emp.setSalary(sal);              System.out.println("\n Salary is incremented \n");

            System.out.println(emp);

        }**else** **if** ((sal >=1500) && (sal <=20000)){

            //incrementing salary 5%

            increment += (sal \* 5)/100;

            sal = sal+increment;

            emp.setSalary(sal);

            System.out.println("\n Salary is incremented \n");

            System.out.println(emp);

        }**else** {

            System.out.println("\n Salary is not incremented \n");

            System.out.println(emp);

  }

    }

}

**Q2) Create an abstract class “order” having members id, description. Create two subclasses**

**“Purchase Order” 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.**

code:-

import

java.io.BufferedReader;impor

tjava.io.IOException;import

java.io.InputStreamReader;a

bstractclass Order{

Stringid,description;

}

class PurchaseOrder extends

Order{String

Customername,Vendorname;public

voidaccept()throwsIOException{

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();Customerna

me=br.readLine();Vendorname=br.read

Line();

}

public void

display(){System.out.pri

ntln("id:"+id);

System.out.println("Description: "+description);System.out.println("Customern

ame: "+Customername);System.out.println("Vendor

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

");

}

}

class SalesOrder extends

Order{StringCustomername,V

endorname;

publicvoidaccept()throwsIOException{

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();Customerna

me=br.readLine();Vendorname=br.read

Line();

}

public void

display(){System.out.pri

ntln("id:"+id);

System.out.println("Description: "+description);System.out.println("Customern

ame: "+Customername);System.out.println("Vendor

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

");

}

}

publicclassMain {

publicstaticvoidmain(String[]args)throwsIOException{

int i;

System.out.println("SelectAnyOne:");

BufferedReaderbr=newBufferedReader(newInputStreamReader(System.in));System.out.println("1.PurchaseOrder");

System.out.println("2.Sales

Order");intch=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].a

ccept();

}

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

{l[i].display();

System.out.println("Objectiscreated");

}

break;

case

2:

System.out.println("Enter the number of sales

orders: ");intm=Integer.parseInt(br.readLine());

SalesOrder [] h=new

SalesOrder[m];for(i=0;i<m;i++)

{

h[i]=new

SalesOrder();h[i].ac

cept();

}

for(i=0;i<m;i+

+){h[i].display()

;

System.out.println("Objectiscreated");

}

break;

}

}

}

Output:- Select Any

One:1.Purchas

e

Order2.SalesO

rder

1

Enter the number of purchase

Orders:2

Enter the id,description,names of customers

and vendors:1

Soap

Rahu

lGajn

i

Enter the id,description,names of customers

and vendors:2

Handwa

shprince

abhishek

id: 1

Description:

SoapCustomernam

e:

RahulVendorname:

Gajni

Object is

createdid: 2

Description:

HandwashCustomer

name:

princeVendorname:

abhishek

Objectiscreated

**Slip 7**

**Q1) Design a class for Bank. Bank Class should support following operations;**

**a. Deposit a certain amount into an account**

* **Withdraw a certain amount from an account**
* **Return a Balance value specifying the amount with details**

code:-

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();

}

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);

sc.close();

}

}

**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**

// Java program to read content from one file

// and write it into another file

// Custom paths for this program

// Reading from - gfgInput.txt

// Writing to - gfgOutput.txt

// Importing input output classes

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

// Class

class GFG {

// Main driver method

public static void main(String[] args)

{

// The file reading process may sometimes give

// IOException

// Try block to check for exceptions

try {

// Creating a FileReader object and

// file to be read is passed as in parameters

// from the local directory of computer

FileReader fr = new FileReader("gfgInput.txt");

// FileReader will open that file from that

// directory, if there is no file found it will

// through an IOException

// Creating a FileWriter object

FileWriter fw = new FileWriter("gfgOutput.txt");

// It will create a new file with name

// "gfgOutput.text", if it is already available,

// then it will open that instead

// Declaring a blank string in which

// whole content of file is to be stored

String str = "";

int i;

// read() method will read the file character by

// character and print it until it end the end

// of the file

// Condition check

// Reading the file using read() method which

// returns -1 at EOF while reading

while ((i = fr.read()) != -1) {

// Storing every character in the string

str += (char)i;

}

// Print and display the string that

// contains file data

System.out.println(str);

// Writing above string data to

// FileWriter object

fw.write(str);

// Closing the file using close() method

// of Reader class which closes the stream &

// release resources that were busy in stream

fr.close();

fw.close();

// Display message

System.out.println(

"File reading and writing both done");

}

// Catch block to handle the exception

catch (IOException e) {

// If there is no file in specified path or

// any other error occurred during runtime

// then it will print IOException

// Display message

System.out.println(

"There are some IOException");

}

}

}

**Slip 8**

**Q1) Create a class Sphere, to calculate the volume and surface area of sphere.**

**(Hint : Surface area=4\*3.14(r\*r), Volume=(4/3)3.14(r\*r\*r))**

// Java program to calculate Volume and

// Surface area of Sphere

class GFG {

// Initializing Value Of PI

static float pi = 3.14159f;

// Function To Calculate Volume Of Sphere

static float volume(float r)

{

float vol;

vol = ((float)4 / (float)3) \* (pi \* r \* r \* r);

return vol;

}

// Function To Calculate Surface Area of Sphere

static float surface\_area(float r) {

float sur\_ar;

sur\_ar = 4 \* pi \* r \* r;

return sur\_ar;

}

// Driver Function

public static void main(String[] args)

{

float radius = 12;

float vol, sur\_area;

// Function Call

vol = volume(radius);

sur\_area = surface\_area(radius);

// Printing Value Of Volume And Surface Area

System.out.println("Volume Of Sphere :" + vol);

System.out.println("Surface Area Of Sphere :" + sur\_area);

}

}

// This code is contributed by Anant Agarwal**.**

**Q2) Design a screen to handle the Mouse Events such as MOUSE\_MOVED and MOUSE\_CLICKED and display the position of the Mouse\_Click in a TextField.**

import java.awt.\*;

import java.awt.event.\*;

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);

}

}

}

class Slip4

{

public static void main(String args[])

{

MyFrame f = new MyFrame("Slip Number 4");

}

}

**Slip 9**

**Q1) Define a “Clock” class that does the following ; a. Accept Hours, Minutes and Seconds**

* **Check the validity of numbers**
* **Set the time to AM/PM mode**

Use the necessary constructors and methods to do the above task

public class Clock {

private int hr; //store hours

private int min; //store minutes

private int sec; //store seconds

// Default constructor

public Clock () {

setClock (0, 0, 0);

}

public Clock (int hours, int minutes, int seconds) {

setClock (hours, minutes, seconds);

}

public void setClock (int hours, int minutes, int seconds) {

if (0 <= hours && hours < 24)

hr = hours;

else

hr = 0;

if (0 <= minutes && minutes < 60)

min = minutes;

else

min = 0;

if (0 <= seconds && seconds < 60)

sec = seconds;

else

sec = 0;

}

public int getHours() {

return hr;

}

public int getMinutes() {

return min;

}

public int getSeconds() {

return sec;

}

public void addClock( Clock secondClock ) {

this.sec += secondClock.getSeconds();

this.min += secondClock.getMinutes();

//add overflow to minutes from seconds

this.min +=(int)(this.sec/60);

//update seconds

this.sec = this.sec % 60;

this.hr += secondClock.getHours();

//add overflow to minutes from seconds

this.hr +=(int)(this.min/60);

//update minutes

this.min = this.min % 60;

//adjust hours

this.hr = this.hr % 24;

}

public void tick(){

this.sec += 1;

//add overflow to minutes from seconds

this.min +=(int)(this.sec/60);

//update seconds

this.sec = this.sec % 60;

//add overflow to minutes from seconds

this.hr +=(int)(this.min/60);

//update minutes

this.min = this.min % 60;

//adjust hours

this.hr = this.hr %24;

}

public void tickDown(){

this.sec -= 1;

if(this.sec <0){

this.sec+=60;

this.min-=1;

}

if(this.min<0){

this.min+=60;

this.hr-=1;

}

if(this.hr<0){

this.hr+=24;

}

}

}

**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.**

import java.util.\*;

interface ProductMarker

{

}

class Product implements ProductMarker

{

int id;

String name;

int cost;

int quantity;

int count;

Product(){

id=0;

name=" ";

cost=0;

quantity=0;

}

Product(int id, String name, int cost, int quantity){

this.id=id;

this.name=name;

this.cost=cost;

this.quantity=quantity;

this.count++;

}

}

public class Products

{

public static void main(String[] args)

{

int count=0;

Scanner a = new Scanner(System.in);

System.out.println("How many product ?");

int number = a.nextInt();

System.out.println("\n");

Product products[] = new Product[number];

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

for(int k=0; k<number; k++)

{

System.out.println("Product Id ");

int id =a.nextInt();

System.out.println("Product name ");

String name = a.next();

System.out.println("Product cost ");

int cost = a.nextInt();

System.out.println("Product qantity ");

int quantity = a.nextInt();

System.out.println("\n");

products[k] = new Product(id, name, cost, quantity);

count++;

}

//Testing for marker interface

if(products[0] instanceof ProductMarker){

System.out.println("Class is using ProductMarker");

}

System.out.println(" Product details\n");

for(Product product:products)

{

System.out.println("Product Id " + product.id);

System.out.println("Product name " + product.name);

System.out.println("Product cost " + product.cost);

System.out.println("Product qantity " + product.quantity);

System.out.println("\n");

}

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

}

}

**SLIP 10**

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

import java.util.Scanner;

public class Exercise13 {

public static void main(String[] args)

{

int i,n;

System.out.print("Input number of terms : ");

Scanner in = new Scanner(System.in);

n = in.nextInt();

for(i=1;i<=n;i++)

{

System.out.println("Number is : " +i+" and cube of " +i+" is : "+(i\*i\*i));

}

}

}

**Q2) Write a program to create a package name student. Define class StudentInfo with method to display information about student such as rollno, class, and percentage. Create another class**

**StudentPer with method to find percentage of the student. Accept student details like rollno, name, class and marks of 6 subject from user.**

import java.io.\*;  
class Student  
{  
int rollno;  
String name;  
int number\_of\_subjects;  
int mark[];

Student(int roll,String stud\_name,int noofsub)throws IOException  
{  
rollno=roll;  
name=stud\_name;  
number\_of\_subjects= noofsub;  
getMarks(noofsub);  
}  
public void getMarks(int nosub ) throws IOException  
{  
mark=new int[nosub];  
BufferedReader br= new BufferedReader (new InputStreamReader(System.in));  
for (int i=0; i<nosub;i++)  
{  
System.out.println(“Enter “+i+”Subject Marks.:=> “);  
mark[i]=Integer.parseInt(br.readLine());  
System.out.println(“”);  
}

}  
public void calculateMarks()  
{  
double percentage=0;  
String grade;  
int tmarks=0;  
for (int i=0;i<mark.length;i++)  
{  
tmarks+=mark[i];  
}  
percentage=tmarks/number\_of\_subjects;  
System.out.println(“Roll Number :=> “+rollno);  
System.out.println(“Name Of Student is :=> “+name);  
System.out.println(“Number Of Subject :=> “+number\_of\_subjects);  
System.out.println(“Percentage Is :=> “+percentage);

if (percentage>=70)  
System.out.println(“Grade Is First Class With Distinction “);  
else if(percentage>=60 && percentage<70)  
System.out.println(“Grade Is First Class”);  
else if(percentage>=50 && percentage<60)  
System.out.println(“Grade Is Second Class”);  
else if(percentage>=40 && percentage<50)  
System.out.println(“Grade Is Pass Class”);  
else  
System.out.println(“You Are Fail”);  
}  
}  
class StudentDemo  
{  
public static void main(String args[])throws IOException  
{  
int rno,no,nostud;  
String name;  
BufferedReader br= new BufferedReader (new InputStreamReader(System.in));  
System.out.println(“Enter How many Students:=> “);  
nostud=Integer.parseInt(br.readLine());  
Student s[]=new Student[nostud];

for(int i=0;i<nostud;i++)  
{  
System.out.println(“Enter Roll Number:=> “);  
rno=Integer.parseInt(br.readLine());  
System.out.println(“Enter Name:=> “);  
name=br.readLine();  
System.out.println(“Enter No of Subject:=> “);  
no=Integer.parseInt(br.readLine());  
s[i]=new Student(rno,name,no);  
}  
for(int i=0;i<nostud;i++)  
{  
s[i].calculateMarks();  
}

}  
}

**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.**

**code:-**

import java.util.\*;

interface Operation{

double PI = 3.142;

void Area(float radius,float height);

void Volume(float radius,float height);

}

class Cylinder implements Operation{

public void Area(float radius,float height){

double area = 2\*PI\*(radius\*height)+2\*PI\*(radius\*radius);

System.out.println("Area of Cylinder: "+area);

}

public void Volume(float radius,float height){

double volume = PI\*(radius\*radius)\*height;

System.out.println("Volume of Cylinder: "+volume);

}

public static void main(String[] args){

Scanner sc = new Scanner(System.in);

System.out.println("Enter radius and height of cylinder:");

float radius = sc.nextFloat();

float height = sc.nextFloat();

Operation ob = new Cylinder();

ob.Area(radius,height);

ob.Volume(radius,height);

sc.close();

}

}

**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.**

// Java code to validate a password

public class PasswordValidator {

// A utility function to check

// whether a password is valid or not

public static boolean isValid(String password)

{

// for checking if password length

// is between 8 and 15

if (!((password.length() >= 8)

&& (password.length() <= 15))) {

return false;

}

// to check space

if (password.contains(" ")) {

return false;

}

if (true) {

int count = 0;

// check digits from 0 to 9

for (int i = 0; i <= 9; i++) {

// to convert int to string

String str1 = Integer.toString(i);

if (password.contains(str1)) {

count = 1;

}

}

if (count == 0) {

return false;

}

}

// for special characters

if (!(password.contains("@") || password.contains("#")

|| password.contains("!") || password.contains("~")

|| password.contains("$") || password.contains("%")

|| password.contains("^") || password.contains("&")

|| password.contains("\*") || password.contains("(")

|| password.contains(")") || password.contains("-")

|| password.contains("+") || password.contains("/")

|| password.contains(":") || password.contains(".")

|| password.contains(", ") || password.contains("<")

|| password.contains(">") || password.contains("?")

|| password.contains("|"))) {

return false;

}

if (true) {

int count = 0;

// checking capital letters

for (int i = 65; i <= 90; i++) {

// type casting

char c = (char)i;

String str1 = Character.toString(c);

if (password.contains(str1)) {

count = 1;

}

}

if (count == 0) {

return false;

}

}

if (true) {

int count = 0;

// checking small letters

for (int i = 97; i <= 122; i++) {

// type casting

char c = (char)i;

String str1 = Character.toString(c);

if (password.contains(str1)) {

count = 1;

}

}

if (count == 0) {

return false;

}

}

// if all conditions fails

return true;

}

// Driver code

public static void main(String[] args)

{

String password1 = "GeeksForGeeks";

if (isValid(password1)) {

System.out.println(password1 + " - Valid Password");

}

else {

System.out.println(password1 + " - Invalid Password!");

}

String password2 = "Geek$ForGeeks7";

if (isValid(password2)) {

System.out.println(password2 + " - Valid Password");

}

else {

System.out.println(password2 + " - Invalid Password!");

}}}

**Slip 12**

**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.**

**code:-**

import java.util.Scanner;

class College

{

int cno;

String cname;

String caddr;

College()

{

Scanner sc = new Scanner(System.in);

System.out.println("Enter the college number:");

cno = sc.nextInt();

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

cname = sc.next();

System.out.println("Enter the college Address:");

caddr = sc.next();

}

void display()

{

System.out.println("College Number :"+cno+" College Name :"+cname+" College Address :"+caddr);

}

}

public class College\_Department extends College // class Department extends College

{

int dno;

String dname;

Scanner sc = new Scanner(System.in);

College\_Department()

{

super();

System.out.println("Enter the department number :");

dno = sc.nextInt();

System.out.println("Enter the department name: ");

dname = sc.next();

}

void display()

{

super.display();

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

}

**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.**

|  |
| --- |
| Simple Calculator |
| |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  | | --- | | 1 | | 4 | | | |  | | --- | | 2 | | 5 | | | | |  | | --- | | 3 | | 6 | | | |  | | --- | | + | | - | | | | 7 |  | | 8 |  | | 9 | | \* | | 0 |  | | . |  | | = | | / | |

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

public class myevent extends Applet implements ActionListener

{

Label   l1,l2,l3;

Button b1,b2,b3,b4,b5;

TextField t1,t2,t3;

public void init()

{

setLayout(null);

l1 = new Label(“Enter the first value”);

l2 = new Label(“Enter the second value”);

l3 = new Label(“Result”);

t1 = new TextField(30);

t2 = new TextField(30);

t3 = new TextField(30);

b1 = new Button(“ADDITION”);

b2 = new Button(“MULTIPLICATION”);

b3 = new Button(“SUBTRACTION”);

b4 = new Button(“DivISION”);

b5 = new Button(“CLEAR”);

 add(t1);

add(t2);

add(t3);

add(l1);

add(l3);

add(l2);

add(b1);

add(b2);

add(b3);

add(b4);

add(b5);

l1.setBounds(50,100,150,30);

l2.setBounds(50,150,150,30);

l3.setBounds(50,200,150,30);

t1.setBounds(250,100,150,30);

t2.setBounds(250,150,150,30);

t3.setBounds(250,200,150,30);

//t3.setVisible(false);

b1.setBounds(230,240,100,30);

b2.setBounds(330,240,100,30);

b3.setBounds(430,240,100,30);

b4.setBounds(530,240,100,30);

b5.setBounds(630,240,100,30);

b1.addActionListener(this);

b2.addActionListener(this);

b3.addActionListener(this);

b4.addActionListener(this);

b5.addActionListener(this);

}

public void actionPerformed(ActionEvent e)

{

if(e.getActionCommand()==”ADDITION”)

{

int a = Integer.parseInt(t1.getText());

int b = Integer.parseInt(t2.getText());

int c = a+b;

t3.setText(String.valueOf(c));

}

if(e.getActionCommand()==”MULTIPLICATION”)

{

int a = Integer.parseInt(t1.getText());

int b = Integer.parseInt(t2.getText());

int c = a\*b;

t3.setText(String.valueOf(c));

}

if(e.getActionCommand()==”SUBTRACTION”)

{

int a = Integer.parseInt(t1.getText());

int b = Integer.parseInt(t2.getText());

int c = a-b;

t3.setText(String.valueOf(c));

}

if(e.getActionCommand()==”DivISION”)

{

int a = Integer.parseInt(t1.getText());

int b = Integer.parseInt(t2.getText());

int c = a/b;

t3.setText(String.valueOf(c));

}

if(e.getActionCommand()==”CLEAR”)

{

t1.setText(“”);

t2.setText(“”);

t3.setText(“”);

}

}

}

/\*<applet code=”myevent” width =1000 height =1000></applet>\*/

**Slip 13**

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

**import** java.io.BufferedReader;

**import** java.io.FileReader;

**public** **class** CountWordFile

{

**public** **static** **void** main(String[] args) **throws** Exception {

        String line;

**int** count = 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**) {

            //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);

        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**

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).**

import java.io.\*;  
class NumberZeroException extends Exception  
{  
    public String toString()  
    {  
        return("Number is 0");  
    }  
}  
  
class PrimeNumber  
{  
    int a;  
    BufferedReader br=new BufferedReader(new InputStreamReader(System.in));  
    PrimeNumber()  
    {  
        try  
        {  
            System.out.println("Enter any integer to check prime ");  
            a=Integer.parseInt(br.readLine());  
            if(a==0)  
                throw new NumberZeroException();  
        }  
        catch(NumberZeroException ex)  
        {  
            System.out.println(ex);  
        }  
        catch(IOException ex1)  
        {  
            System.out.println("Enter proper number");  
        }  
    }  
  
    public void prime()  
    {  
        int cnt=0;  
        for(int i=2;i<=a/2;i++)  
            if(a%i==0)  
            {  
                cnt++;  
                break;  
            }  
        if(cnt==0)  
            System.out.println(a+" Number is prime");  
        else  
            System.out.println(a+" Number is not prime");  
    }  
    public static void main(String args[])  
    {  
        PrimeNumber pn=new PrimeNumber();  
        pn.prime();  
    }  
}

**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.**

**code:-**

**packageAssignment2.SY;**

**import**

**java.io.BufferedReader;im**

**portjava.io.\*;**

**public class**

**SYClass**

**{publicintct,mt,et;**

**publicvoidget()throwsIOException{**

**System.out.println("Entermarksofstudentsforcomputer,mathsandelectronicssubject**

**outof200");**

**BufferedReaderbr=newBufferedReader(newInputStreamReader(System.in));ct=Inte**

**ger.parseInt(br.readLine());**

**mt=Integer.parseInt(br.readLi**

**ne());et=Integer.parseInt(br.r**

**eadLine());**

**}**

**}**

**Program2:**

**package**

**Assignment2.TY;impo**

**rtjava.io.\*;**

**publicclassTYClass{**

**publicinttm,pm;**

**publicvoidget()throwsIOException{**

**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());**

**}**

**}**

**Program3:**

**package**

**Assignment2;import**

**Assignment2.SY.\*;imp**

**ort**

**Assignment2.TY.\*;imp**

**ortjava.io.\*;**

**class**

**StudentInfo{in**

**t rollno;**

**String**

**name,grade;public**

**float**

**gt,tyt,syt;publicfl**

**oatper;**

**publicvoidget()throwsIOException{**

**System.out.println("Enter roll number and name of the student: ");BufferedReaderbr=newBufferedReader(newInputStreamReader(S**

**ystem.in));rollno=Integer.parseInt(br.readLine());**

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

**}**

**}**

**publicclassStudentMarks{**

**publicstaticvoidmain(String[]args)throwsIOException{**

**BufferedReaderbr=newBufferedReader(newInputStreamReader(System.in));System. out.println("Enter thenumberof students:");**

**intn=Integer.parseInt(br.readLi**

**ne());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].ty**

**t;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";elseif(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\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**.

import java.io.\*;

import java.util.\*;

class copy{

public static void main(String arg[]) throws Exception {

Scanner sc = new Scanner(System.in);

System.out.print("source file name :\n");

String file1 = sc.next();

System.out.print("destination file name :\n");

String file2 = sc.next();

FileReader fin = new FileReader(file1);

FileWriter fout = new FileWriter(file2, true);

int c;

while ((c = fin.read()) != -1) {

fout.write(c);

}

System.out.println("Copy finish...");

fin.close();

fout.close();

}

}

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

// Parameterized Constructor Example in Java

import java.io.\*;

class parameterizedConstructor {

// fields of the class

String name;

int regestrationNumber;

// creating a parameterized constructor so that we can

// initialize the value of the class

parameterizedConstructor(String name,

int regestrationNumber)

{

System.out.println("constructor call");

this.name = name;

this.regestrationNumber = regestrationNumber;

}

}

class GFG {

public static void main(String[] args)

{

// creating our first object

parameterizedConstructor obj1

= new parameterizedConstructor("Nilesh",

2021806);

System.out.println("Name of the student "

+ obj1.name);

System.out.println("Registration Number "

+ obj1.regestrationNumber);

// creating second object

parameterizedConstructor obj2

= new parameterizedConstructor("Bhaskar",

2021807);

System.out.println("Name of the student "

+ obj2.name);

System.out.println("Registration Number "

+ obj2.regestrationNumber);

}

}

Slip 16

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

code:-

import java.util.Scanner;

interface Square{

public int Square(int n);

}

class findingcube implements FindingCube{

public int Square(int n){

return (n\*n);

}

}

public static void main(String args[]){

Square cc = new Square();

System.out.println("Enter a number ::");

Scanner sc = new Scanner(System.in);

int num=sc.nextInt();

Square = cc.Square(num);

System.out.println("Cube of the given number is "+Square);

}

}

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



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 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.**

#include<iostream.h>  
#include<conio.h>  
class customer  
{  
char name[20];  
char pno[10];  
public:  
void get()  
{  
cout<<“\nEnter the Name & Phone no.\n”;  
cin>>name>>pno;  
}  
void disp\_c()  
{  
cout<<name<<“\t”<<pno<<“\t\t”;  
}  
};  
class depositor:public customer  
{

int accno;  
float bal;  
public:  
void get\_d()  
{  
cout<<“\nEnter the A/c no. & Balance: \n”;  
cin>>accno>>bal;  
}  
void disp\_d()  
{  
cout<<“\t”<<accno<<“\t”<<“\t\t”<<bal<<endl;  
}  
};  
class borrower:public depositor  
{  
int lno;  
float lamt;  
public:  
void get\_b()  
{  
get();  
get\_d();  
cout<<“\nEnter the loan no. & loan amount: \n”;  
cin>>lno>>lamt;  
}  
void display()  
{  
disp\_c();  
disp\_d();  
cout<<“\t”<<lno<<“\t\t”<<lamt<<endl;  
}  
};  
void main()  
{  
int n,i;  
borrower b[10];  
clrscr();  
cout<<“\nHow many customer: \n”;  
cin>>n;  
for(i=0;i<n;i++)  
{  
b[i].get\_b();  
}  
cout<<“\nThe Detail’s of Customer’s: \n”;  
cout<<“Name\tPNo.\tA/C no.\t\tBalance\t\tLoan no.\tLoan amt\n”;  
cout<<“============================================\n”;  
for(i=0;i<n;i++)  
{  
b[i].display();  
}  
getch();  
}

**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**

**code:-**

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

class TextBoxes extends JFrame implements ActionListener

{

String op;

String ans,ans1,rev;

JTextField t1,t2,t3;

JButton b1,b2;

JPanel p1,p2;

public TextBoxes()

{

t1=new JTextField(20);

t2=new JTextField(20);

t3=new JTextField(20);

b1=new JButton("Concatenation");

b2=new JButton("Reverse");

p1=new JPanel();

p1.setLayout(new GridLayout(3,1));

p1.add(t1);

p1.add(t2);

p1.add(t3);

p2=new JPanel();

p2.setLayout(new GridLayout(2,1));

p2.add(b1);

p2.add(b2);

setLayout(new GridLayout(2,1));

add(p1);

add(p2);

b1.addActionListener(this);

b2.addActionListener(this);

setBounds(100,100,500,500);

setVisible(true);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

}

public void actionPerformed(ActionEvent ae)

{

JButton b=(JButton)ae.getSource();

String s1=t1.getText();

String s2=t2.getText();

if(b==b1)

{

ans=s1+s2;

t3.setText(ans);

}

else if(b==b2)

{

if(t3.getText()!="")

{

StringBuffer sb=new StringBuffer(ans);

sb.reverse();

t3.setText(sb.toString());

}

else{

ans1="STRING NOT AVAILABLE";

t3.setText(ans1);

}

}

}

public static void main(String[] args)

{

new TextBoxes();

}

}

**Slip 18**

**Q1) Write a program to implement Border Layout Manager**.

**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();

}

}

**Q2Define a class CricketPlayer (name,no\_of\_innings,no\_of\_times\_notout, totatruns, bat\_avg). Create an array of n player objects. Calculate the batting average for each player using static method avg(). Define a static sort method which sorts the array on the basis of average.**

**Display the player details in sorted order.**

import java.io.\*;

class Cricket {

String name;

int inning, tofnotout, totalruns;

float batavg;

public Cricket(){

name=null;

inning=0;

tofnotout=0;

totalruns=0;

batavg=0;

}

public void get() throws IOException{

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

System.out.println("Enter the name, no of innings, no of times not out, total runs: ");

name=br.readLine();

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

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

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

}

public void put(){

System.out.println("Name="+name);

System.out.println("no of innings="+inning);

System.out.println("no times notout="+tofnotout);

System.out.println("total runs="+totalruns);

System.out.println("bat avg="+batavg);

}

static void avg(int n, Cricket c[]){

try{

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

c[i].batavg=c[i].totalruns/c[i].inning;

}

}catch(ArithmeticException e){

System.out.println("Invalid arg");

}

}

static void sort(int n, Cricket c[]){

String temp1;

int temp2,temp3,temp4;

float temp5;

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

for(int j=i+1;j<n;j++){

if(c[i].batavg<c[j].batavg){

temp1=c[i].name;

c[i].name=c[j].name;

c[j].name=temp1;

temp2=c[i].inning;

c[i].inning=c[j].inning;

c[j].inning=temp2;

temp3=c[i].tofnotout;

c[i].tofnotout=c[j].tofnotout;

c[j].tofnotout=temp3;

temp4=c[i].totalruns;

c[i].totalruns=c[j].totalruns;

c[j].totalruns=temp4;

temp5=c[i].batavg;

c[i].batavg=c[j].batavg;

c[j].batavg=temp5;

}

}

}

}

}

public class a4sa1 {

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

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

System.out.println("Enter the limit:");

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

Cricket c[]=new Cricket[n];

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

c[i]=new Cricket();

c[i].get();

}

Cricket.avg(n,c);

Cricket.sort(n, c);

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

c[i].put();

}

}

}

**Slip 19**

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

// Java Program to Find the Sum of Diagonals of a Matrix

// Importing input output classes

import java.io.\*;

// Main Class

public class GFG {

// To calculate Sum of Diagonals

static void Sum\_of\_Diagonals1(int[][] matrix, int N)

{

// Declaring and initializing two variables to zero

// initially for primary and secondary diagonal

// count

int Pd = 0, Sd = 0;

// Two Nested for loops for iteration over a matrix

// Outer loop for rows

for (int k = 0; k < N; k++) {

// Inner loop for columns

for (int l = 0; l < N; l++) {

// Condition for the principal

// diagonal

if (k == l)

Pd += matrix[k][l];

// Condition for the secondary diagonal

if ((k + l) == (N - 1))

Sd += matrix[k][l];

}

}

// Print and display the sum of primary diagonal

System.out.println("Sum of Principal Diagonal:"

+ Pd);

// Print and display the sum of secondary diagonal

System.out.println("Sum of Secondary Diagonal:"

+ Sd);

}

// Main driver method

static public void main(String[] args)

{

// Input integer array

// Custom entries in an array

int[][] b = { { 8, 2, 13, 4 },

{ 9, 16, 17, 8 },

{ 1, 22, 3, 14 },

{ 15, 6, 17, 8 } };

// Passing the array as an argument to the

// function defined above

Sum\_of\_Diagonals1(b, 4);

}

}

**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.**

[20 marks]

**code:-**

import javax.swing.\*;  
import java.awt.event.\*;  
public class JCombo\_Box  
{  
 JFrame f;  
 JCombo\_Box()  
 {  
 f = new JFrame("ComboBox Example");  
 JTextField j = new JTextField(50);  
 final JLabel label = new JLabel();  
 label.setHorizontalAlignment(JLabel.*CENTER*);  
 label.setSize(400,100);  
 JButton b=new JButton("Show");  
 b.setBounds(200,100,75,20);  
 String languages[]={"C","C++","C#","Java","PHP"};  
 final JComboBox cb=new JComboBox(languages);  
 cb.setBounds(50, 100,90,20);  
 f.add(cb); f.add(label); f.add(b);f.add(j);  
 f.setLayout(null);  
 f.setSize(350,350);  
 f.setVisible(true);  
 b.addActionListener(new ActionListener()

{  
 public void actionPerformed(ActionEvent e)

{  
 String data = "Programming language Selected: "  
 + cb.getItemAt(cb.getSelectedIndex());  
 label.setText(data);  
 }  
 });  
 }  
 public static void main(String[] args) {  
 new JCombo\_Box();  
 }  
}

**SLIP 20**

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

import java.io.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**

import java.util.Scanner;

public class JavaProgram

{

public static void main(String args[])

{

int first, second, add, subtract, multiply;

float devide;

Scanner scanner = new Scanner(System.in);

System.out.print("Enter Two Numbers : ");

first = scanner.nextInt();

second = scanner.nextInt();

add = first + second;

subtract = first - second;

multiply = first \* second;

devide = (float) first / second;

System.out.println("Sum = " + add);

System.out.println("Difference = " + subtract);

System.out.println("Multiplication = " + multiply);

System.out.println("Division = " + devide);

}

}

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.**

**code:-**

import java.util.Scanner;  
class InvalidDateException extends Exception  
{  
}  
class MyDate  
{  
 int day,mon,yr;  
 void accept(int d,int m,int y)  
 {  
 day = d;  
 mon = m;  
 yr = y;  
 }  
 void display()  
 {  
 System.*out*.println("Date is valid : "+day+"/"+mon+"/"+yr);  
 }  
}  
class date  
{  
 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 mon = sc.nextInt();  
 int yr = sc.nextInt();  
 int flag=0;  
 try  
 {  
 if(mon<=0 || mon>12)  
 throw new InvalidDateException();  
 else  
 {  
 if(mon==1 || mon==3 || mon==5 || mon==7 || mon==8 || mon==10 || mon == 12)  
 {  
 if(day>=1 && day <=31)  
 flag=1;  
 else  
 throw new InvalidDateException();  
 }  
 else if (mon==2)  
 {  
 if(yr%4==0)  
 {  
 if(day>=1 && day<=29)  
 flag=1;  
 else  
 throw new InvalidDateException();  
 }  
 else  
 {  
 if(day>=1 && day<=28)  
 flag=1;  
 else  
 throw new InvalidDateException();  
 }  
 }  
 else  
 {  
 if(mon==4 || mon == 6 || mon== 9 || mon==11)  
 {  
 if(day>=1 && day <=30)  
 flag=1;  
 else throw new InvalidDateException();  
 }  
 }  
 }  
 if(flag== 1)  
 {  
 MyDate dt = new MyDate();  
 dt.accept(day,mon,yr);  
 dt.display();  
 }  
 }  
 catch (InvalidDateException mm)  
 {  
 System.*out*.println("Invalid Date");  
 }  
 }  
}

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

import java.util.Scanner;

public class Employee {

int id;

String name;

String deptname;

float salary;

static int numberofobjects=0;

Employee(){

id=0;

name="";

deptname="";

salary=0;

}

Employee(int id,String name,String deptname,float salary ){

this.id=id;

this.name=name;

this.deptname=deptname;

this.salary=salary;

numberofobjects++;

}

public void display(){

System.out.println("Employee Id :"+id);

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

System.out.println("Employee Department: "+deptname);

System.out.println("Employee Salary :"+salary);

}

public static void main(String[] args){

int n=0;

Scanner sc=new Scanner(System.in);

System.out.print("How many employees you want to enter :");

n=sc.nextInt();

Employee[] ob=new Employee[n];

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

sc= new Scanner(System.in);

System.out.println("Enter Id of employee "+(i+1)+" :");

int id=sc.nextInt();

System.out.println("Enter Name of employee "+(i+1)+" :");

sc.nextLine();

String name= sc.nextLine();

System.out.println("Enter dept name of employee "+(i+1)+" :");

String deptname=sc.nextLine();

System.out.println("Enter salary of employee "+(i+1)+" :");

float salary = sc.nextFloat();

ob[i]=new Employee(id,name,deptname,salary);

System.out.println("\nNumber of Objects : "+numberofobjects);

}

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

{

ob[i].display();

}

}

}

Slip22

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

import java.util.\*;

abstract class shape

{

int x,y;

abstract void area(double x,double y);

}

class Rectangle extends shape

{

void area(double x,double y)

{

System.out.println("Area of rectangle is :"+(x\*y));

}

}

class Circle extends shape

{

void area(double x,double y)

{

System.out.println("Area of circle is :"+(3.14\*x\*x));

}

}

class Triangle extends shape

{

void area(double x,double y)

{

System.out.println("Area of triangle is :"+(0.5\*x\*y));

}

}

public class AbstactDDemo

{

public static void main(String[] args)

{

Rectangle r=new Rectangle();

r.area(2,5);

Circle c=new Circle();

c.area(5,5);

Triangle t=new Triangle();

t.area(2,5);

}

}

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

import javax.swing.event.\*;

import java.awt.event.\*;

import java.awt.\*;

import javax.swing.JApplet;

public class mouseevnts extends JApplet implements MouseListener

{

private int x,y;

private String event;

public void init()

{

setLayout(new FlowLayout());

x=-1;

addMouseListener(this);

}

public void paint(Graphics g)

{

super.paint(g);

g.drawRect(0,0,getWidth(),getHeight());

if(x!=1)

{

g.drawString("Mouseevent is"+event+"("+x+","+y+")", 10,50);

}

}

public void mousePressed(MouseEvent e)

{

x=e.getX();

y=e.getY();

event="pressed";

repaint();

}

public void mouseClicked(MouseEvent e)

{

x=e.getX();

y=e.getY();

event="clicked";

repaint();

}

public void mouseReleased(MouseEvent e)

{

x=e.getX();

y=e.getY();

event="Reeleased";

repaint();

}

public void mouseExited(MouseEvent e)

{

x=e.getX();

y=e.getY();

event="Exited";

repaint();

}

public void mouseEntered(MouseEvent e)

{

x=e.getX();

y=e.getY();

event="Entered";

repaint();

}

}

**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.

**code:-**

public class Ass1\_SetB\_b  
 {  
 private int x;  
 Ass1\_SetB\_b()  
 {  
 x = 0;  
 }  
 Ass1\_SetB\_b(int x)  
 {  
 this.x = x;  
 }  
 void isZero()  
 {  
 if(x==0)  
 System.*out*.println("The number is Zero");  
 }  
 void isPositive()  
 {  
 if(x>0)  
 System.*out*.println("The number is Positive");  
 }  
 void isNegative()  
 {  
 if(x<0)  
 System.*out*.println("The number is Negative");  
 }  
 void isOdd()  
 {  
 if(x%2!=0)  
 System.*out*.println("The number is Odd");  
 }  
 void isEven()  
 {  
 if(x%2==0)  
 System.*out*.println("The number is Even");  
 }  
 public static void main(String [] args) throws ArrayIndexOutOfBoundsException  
 {  
 int num=Integer.*parseInt*(args[0]);  
 Ass1\_SetB\_b obj=new Ass1\_SetB\_b(num);  
 obj.isNegative();  
 obj.isPositive();  
 obj.isEven();  
 obj.isOdd();  
 obj.isZero();   
 } }

**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.**

**code:-**

import java.awt.BorderLayout;

import java.awt.EventQueue;

import javax.swing.JFrame; import javax.swing JPanel;

import javax.swing.border.Empty Border:

import javax.swing II abel;

import javax.swing.JTextField; import javax.swing.JButton;

import java.awt.event.ActionListener;

import java.awt.event.ActionEvent;

public class CurrConv extends JFrame(

private JPanel contentPane;

private ITextField txtSgd;

private JTextField txtUsd;

private ITextField txtFuro;

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable(){

public void run() { Lry (

CurrConv frame = new CurrCony(); frame.setVisible(true);

} catch (Exception e) ( e.printStackTrace();

}

}

});

}

public CurrConv() {

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(100, 100, 450, 300); contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(null):

JLabel IblSingapore Dollars = new JLabel("Singapore Dollars");

IblSingaporeDollars.setBounds(24, 38, 116, 14),

contentPane.add(blSingapore Dollars); JLabel IblUsDollars new JLabel('US Dollars");

IblUsDollars.setBounds(24, 74, 116, 14);

contentPane.add(lblUsDollars);

JLabel IblEuros = new JLabel("Euros");

IblEuros.setBounds(24, 112, 116, 14),

contentPane.add(IblEuros);

txtSgd = new JTextField();

txtSgd.addActionListener(new ActionListener() {

public void actionPerformed (ActionEvent e) {

String strSgd = txtSgd.getText(); double dbSgd = Double.valueOf(strSgd);

double resUsd = dbSgd/1.41;

double resEuro dbSgd \* 0.65;

txtUsd.setText(String.valueOf(resUsd));

txtEuro.setText(String.valueOf(resEuro));

txtSgd.setBounds(212, 35, 86, 20);

contentPane.add(txtSgd); txtSgd.setColumns(10);

txtUsd = new JTextField(); txtUsd.addActionListener(new ActionListener() { public void actionPerformed(ActionEvent e) {

double usd Double.valueOf(txtUst.getText()); txtSgd.set l'ext(String.valueOf(usd\* 1.41});

txtum.setText(String.valueOf(usd"0.92));

}

});

txtUsd.setColumns(10);

txtUsd.setBounds(212, 71, 86, 20);

contentPane.add(txtUsd):

txtEuro new JTextField(): txtEuro.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) | double euro Double.valueOf(txtEuro.getText());

txtSgd.setText(String.valueOf(euro / 0.65));

txtsd.setText(String.valueOf(euro / 0.92));

}

});

txtEuro.setColumns(10); LxIEuro.setBounds(212, 109, 86, 20);

contentPane.add(txtEuro);

}

}

**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.**

abstract class bank  
{  
 abstract void getBalance();  
}  
class A extends bank  
{  
 int rs100;  
 A()  
 {  
 rs100 = 100;  
 }  
 void getBalance()  
 {  
 System.*out*.println("Deposited Rupees -bank A "+rs100);  
 }  
}  
class B extends bank  
{  
 int rs200;  
 B()  
 {  
 rs200 = 200;  
 }  
 void getBalance()  
 {  
 System.*out*.println("Deposited Rupees -bank A "+rs200);  
 }  
}  
class C extends bank  
{  
 int rs300;  
 C()  
 {  
 rs300 = 300;  
 }  
 void getBalance()  
 {  
 System.*out*.println("Deposited Rupees -bank A "+rs300);  
 }  
}  
public class AbstractBank

{  
 public static void main(String[] args) {  
 A a1 = new A();  
 a1.getBalance();  
 B b1 = new B();  
 b1.getBalance();  
 C c1 = new C();  
 c1.getBalance();  
 }  
}

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

**code:-**

include<stdio.h>

#include<graphics.h>

#include<conio.h>

int main(){

int gd = DETECT,gm;

int x ,y;

initgraph(&gd, &gm, "C:\\TC\\BGI");

/\* Initialize center of circle with center of screen \*/

x = getmaxx()/2;

y = getmaxy()/2;

outtextxy(240, 50, "Concentric Circles");

/\* Draw circles on screen \*/

setcolor(RED);

circle(x, y, 30);

setcolor(GREEN);

circle(x, y, 50);

setcolor(YELLOW);

circle(x, y, 70);

setcolor(BLUE);

circle(x, y, 90);

getch();

closegraph();

return 0;

}

**Slip25**

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

**code:-**

import java.io.\* ;

class Except

{

public static void main(String args[])throws Exception

{

InputStreamReader r=new InputStreamReader(System.in);

BufferedReader br=new BufferedReader(r);

System.out.println(“Enter name:”);

String name = br.readLine();

System.out.println(“Enter roll no.:”);

String number=br.readLine();

System.out.println(“Enter marks:”);

String marks=br.readLine();

System.out.println(“name:”+name);

System.out.println(“Roll No.:”+number);

System.out.println(“Marks:”+marks);

}

}

**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.**

**code:-**

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.**

import java.util.Scanner;  
 public class ItemClass

{  
 int item\_number;  
 String item\_name;  
 int item\_price;  
 static int *numberofobjects*=0;  
 ItemClass()  
 {  
 item\_number=0;  
 item\_name="";  
 item\_price=0;  
 }  
 ItemClass(int item\_number,String item\_name,int item\_price)  
 {  
 this.item\_number=item\_number;  
 this.item\_name=item\_name;  
 this.item\_price=item\_price;  
 *numberofobjects*++;  
 }  
 public void display(){  
 System.*out*.println("ItemClass item\_number :"+item\_number);  
 System.*out*.println("ItemClass item\_name: "+item\_name);  
 System.*out*.println("ItemClass price: "+item\_price);  
 }  
 public static void main(String[] args){  
 int n=0;  
 Scanner sc=new Scanner(System.*in*);  
 System.*out*.print("How many Items you want to enter :");  
 n=sc.nextInt();  
 ItemClass[] ob = new ItemClass[n];  
 for(int i=0;i<n;i++)  
 {  
 sc= new Scanner(System.*in*);  
 System.*out*.println("Enter item\_number of item "+(i+1)+" :");  
 int item\_number=sc.nextInt();  
 System.*out*.println("Enter item\_name of item "+(i+1)+" :");  
 String item\_name= sc.next();  
 System.*out*.println("Enter Item price of item "+(i+1)+" :");  
 int item\_price=sc.nextInt();  
 ob[i] = new ItemClass(item\_number,item\_name,item\_price);  
 System.*out*.println("\nNumber of Objects : "+*numberofobjects*);  
 }  
 for(int i=0;i<n;i++)  
 {  
 ob[i].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.**

**code:-**

import java.awt.BorderLayout;

import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.border.EmptyBorder;

import javax.swing.JLabel;

import javax.swing.JTextField;

import javax.swing.JButton;

import java.awt.event.ActionListener;

import java.awt.event.ActionEvent;

public class CurrConv extends JFrame {

private JPanel contentPane;

private JTextField txtSgd;

private JTextField txtUsd;

private JTextField txtEuro;

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

CurrConv frame = new CurrConv ();

frame.setVisible(true);

}

catch (Exception e) {

e.printStackTrace();

}}

});

}

/\*\*\* Create the frame.\*/

public CurrConv() {

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(100, 100, 450, 300);

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(null);

JLabel lblSingaporeDollars = new JLabel("Singapore Dollars");

lblSingaporeDollars.setBounds(24, 38, 116, 14);

contentPane.add(lblSingaporeDollars);

JLabel lblUsDollars = new JLabel("US Dollars");

lblUsDollars.setBounds(24, 74, 116, 14);

contentPane.add(lblUsDollars);

JLabel lblEuros = new JLabel("Euros");

lblEuros.setBounds(24, 112, 116, 14);

contentPane.add(lblEuros);

txtSgd = new JTextField();

txtSgd.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

String strSgd = txtSgd.getText();

double dbSgd = Double.valueOf(strSgd);

double resUsd = dbSgd / 1.41;

double resEuro = dbSgd \* 0.65;

txtUsd.setText(String.valueOf(resUsd));

txtEuro.setText(String.valueOf(resEuro));

}

});

txtSgd.setBounds(212, 35, 86, 20);

contentPane.add(txtSgd);

txtSgd.setColumns(10);

txtUsd = new JTextField();

txtUsd.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

double usd = Double.valueOf(txtUsd.getText());

txtSgd.setText(String.valueOf(usd\*1.41));

txtEuro.setText(String.valueOf(usd\*0.92));

}

});

txtUsd.setColumns(10);

txtUsd.setBounds(212, 71, 86, 20);

contentPane.add(txtUsd);

txtEuro = new JTextField();

txtEuro.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

double euro = Double.valueOf(txtEuro.getText());

txtSgd.setText(String.valueOf(euro / 0.65));

txtUsd.setText(String.valueOf(euro / 0.92));

}

});

txtEuro.setColumns(10);

txtEuro.setBounds(212, 109, 86, 20);

contentPane.add(txtEuro);

}

}

**Slip 27**

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

import java.util.\*;

import java.io.\*;

class Employee

{

String nm;

int id;

float sal;

void accept()

{

System.out.println("Enter id & name");

Scanner s=new Scanner(System.in);

id=s.nextInt();

nm=s.next();

}

float getsalary()

{

System.out.println("Enter salary");

Scanner s=new Scanner(System.in);

sal=s.nextFloat();

return sal;

}

}

class Manager extends Employee

{

int ta,hr;

float s1;

float getsalary()

{

super.accept();

s1=super.getsalary();

System.out.println("Enter travelling allownces & house rent");

Scanner s=new Scanner(System.in);

ta=s.nextInt();

hr=s.nextInt();

System.out.println("Salary after adding="+(s1+ta+hr));

return s1+ta+hr;

}

void display()

{

float a=getsalary();

}

}

class Slip27

{

public static void main(String a[])

{

Manager ob=new Manager();

ob.display();

}

}

**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: If is a directory, delete all text files in that directory, Confirm delete operation from [10 marks] user before deleting text files. Also, display a count showing the number of files deleted. if any, from the directory If is a file display various details of that file.**

**code:-**

import java.io.\*;

public class Test {

public static void main(String[] args)

{

File file

= new File("C:\\Users\\Mayank\\Desktop\\1.txt");

if (file.delete()) {

System.out.println("File deleted successfully");

}

else {

System.out.println("Failed to delete the file");

}

}

**}**

**Slip 28**

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

**Ans:-**

import java.io.\*;

class filedemo

{

public static void p(String str)

{

System.out.println(str);

}

public static void analyze(String s)

File f=new File(s);

if(f.exists())

{

p(f.getName()+" is a file");

p(f.canRead()?" is readable":" is not readable");

p(f.canWrite()?" is writable":" is not writable");

p("Filesize:"+f.length()+" bytes");

p("File last mdified:"+f.lastModified());

}

if(f.isDirectory())

{

p(f.getName()+" is directory");

p("List of files");

String dir[]=f.list();

for(int i=0;i<dir.length;i++)

p(dir[i]);

}

}

}

public class FileDetails

{

public static void main(String rr[])throws IOException

{

filedemo fd=new filedemo();

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

System.out.println("Enter the file name:");

String s=br.readLine();

fd.analyze(s);

}

}

**Q2)Write a program called Swing TemperatureConverter 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 (eg, I decimal place), use the static method String format(), which has the same form as printf(). For example, String format("%.11, 1.234) returns String "1.2".**

**Ans:-**

import java.awt.BorderLayout;

import java.awt.GridLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.BorderFactory;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JPanel;

import javax.swing.JTextField;

import javax.swing.SwingConstants;

import javax.swing.UIManager;

// This example demonstrates the use of JButton, JTextField and JLabel.

public class CelsiusConverter implements ActionListener {

JFrame converterFrame;

JPanel converterPanel;

JTextField tempCelsius;

JLabel celsiusLabel, fahrenheitLabel;

JButton convertTemp;

// Constructor

public CelsiusConverter() {

// Create the frame and container.

converterFrame = new JFrame("Convert Celsius to Fahrenheit");

converterPanel = new JPanel();

converterPanel.setLayout(new GridLayout(2, 2));

// Add the widgets.

addWidgets();

// Add the panel to the frame.

converterFrame.getContentPane()

.add(converterPanel, BorderLayout.CENTER);

// Exit when the window is closed.

converterFrame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

// Show the converter.

converterFrame.pack();

converterFrame.setVisible(true);

}

// Create and add the widgets for converter.

private void addWidgets() {

// Create widgets.

tempCelsius = new JTextField(2);

celsiusLabel = new JLabel("Celsius", SwingConstants.LEFT);

convertTemp = new JButton("Convert...");

fahrenheitLabel = new JLabel("Fahrenheit", SwingConstants.LEFT);

// Listen to events from Convert button.

convertTemp.addActionListener(this);

// Add widgets to container.

converterPanel.add(tempCelsius);

converterPanel.add(celsiusLabel);

converterPanel.add(convertTemp);

converterPanel.add(fahrenheitLabel);

celsiusLabel.setBorder(BorderFactory.createEmptyBorder(5, 5, 5, 5));

fahrenheitLabel.setBorder(BorderFactory.createEmptyBorder(5, 5, 5, 5));

}

// Implementation of ActionListener interface.

public void actionPerformed(ActionEvent event) {

// Parse degrees Celsius as a double and convert to Fahrenheit.

int tempFahr = (int) ((Double.parseDouble(tempCelsius.getText())) \* 1.8 + 32);

fahrenheitLabel.setText(tempFahr + " Fahrenheit");

}

// main method

public static void main(String[] args) {

// Set the look and feel.

try {

UIManager.setLookAndFeel(UIManager

.getCrossPlatformLookAndFeelClassName());

} catch (Exception e) {

}

CelsiusConverter converter = new CelsiusConverter();

}

}

**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.**

**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.**

**code:-**

import java.io.\*;

class Vehicle{

String company;

double price;

public void accept() throws IOException{

System.out.println("Enter the Company and price of the Vehicle: ");

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

company=br.readLine();

price=Double.parseDouble(br.readLine());

}

public void display(){

System.out.println("Company: "+company+" Price: "+price);

}

}

class LightMotorVehicle extends Vehicle{

double mileage;

public void accept() throws IOException{

super.accept();

System.out.println("Enter the mileage of the vehicle: ");

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

mileage=Double.parseDouble(br.readLine());

}

public void display(){

super.display();

System.out.println("Mileage: "+mileage);

}

}

class HeavyMotorVehicle extends Vehicle{

double captons;

public void accept() throws IOException{

super.accept();

System.out.println("Enter the capacity of vehicle in tons: ");

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

captons=Double.parseDouble(br.readLine());

}

public void display(){

super.display();

System.out.println("Capacity in tons: "+captons);

}

}

public class sa3 {

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

int i;

System.out.println("Enter the type of vehicle: ");

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

System.out.println("1.Light Vehicle");

System.out.println("2.Heavy Vehicle");

int ch=Integer.parseInt(br.readLine());

switch(ch){

case 1:

System.out.println("Enter the number of Light vehicles: ");

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

LightMotorVehicle [] l=new LightMotorVehicle[n];

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

l[i]=new LightMotorVehicle();

l[i].accept();

}

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

l[i].display();

}

break;

case 2:

System.out.println("Enter the number of Heavy vehicles: ");

int m=Integer.parseInt(br.readLine());

HeavyMotorVehicle [] h=new HeavyMotorVehicle[m];

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

h[i]=new HeavyMotorVehicle();

h[i].accept();

}

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

h[i].display();

}

break;

}

}

}

**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).**

**code:-**

**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.**

code:-

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

/\*

<applet code="DivApplet" width=350 height=300>

</applet>

\*/

public class DivApplet extends JApplet implements ActionListener{

JTextField number1,number2,result;

JButton divide;

public void init(){

try {

SwingUtilities.invokeAndWait(

new Runnable() {

public void run() {

makeGUI();

}

});

}

catch (Exception exc) {

System.out.println("Can't create because of " + exc);

}

}

private void makeGUI(){

setLayout(new FlowLayout());

Label number1p = new Label("Number1: ",Label.RIGHT);

Label number2p = new Label("Number2: ",Label.RIGHT);

number1= new JTextField(20);

number2 = new JTextField(20);

result = new JTextField(20);

divide = new JButton("Divide");

add(number1p);

add(number1);

add(number2p);

add(number2);

add(result);

add(divide);

divide.addActionListener(this);

}

public void actionPerformed(ActionEvent e){

String snumber1,snumber2;

snumber1 = number1.getText();

snumber2 = number2.getText();

try{

int number1 = Integer.parseInt(snumber1);

int number2 = Integer.parseInt(snumber2);

if(number2==0)

JOptionPane.showMessageDialog(null, "Division by zero not defined.");

else{

double r = (double)number1/number2;

result.setText(((Double)r).toString());

}

}

catch(NumberFormatException ne)

{

JOptionPane.showMessageDialog(null,"Enter a number");

}

}

}