**OOP LAB PROGRAMS**

1. a) Design, Develop and Implement a Java program to calculate gross salary net salary taking the following data

DA=40% of basic

HRA=20% of basic

CCA=Rs250/-

PF=10%of basic

PT=Rs100/-

Income tax = 10% of gross

Gross income: Basic + DA + HRA + CCA

Deductions = PF+PT+IT

Net income = Gross income – Deductions

import java.util.Scanner;

public class EmployeeSalary {

public static void main(String[] args) {

String name,id;

double bSalary, DA, HRA, PF, IT,grossIncome, netIncome, deductions;

double CCA = 250, PT = 100;

Scanner s = new Scanner(System.in);

System.out.println("Enter name of the employee");

name = s.nextLine();

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

id = s.nextLine();

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

bSalary = s.nextDouble();

DA = (0.4)\*bSalary;

HRA = (0.2)\*bSalary;

PF = (0.1)\*bSalary;

grossIncome = bSalary + DA + HRA+CCA;

IT = (0.1)\*grossIncome;

deductions = PF+PT+IT;

netIncome = grossIncome - deductions;

System.out.println("The Gross income of employee "+name+" with ID "+id+" is "+grossIncome);

System.out.println("The Net income of employee "+name+" with ID "+id+" is "+netIncome);

s.close();

}

}

1) b) Design, Develop and Implement a Java program that prints all real solutions to the quadratic equation ax2 + bx + c = 0. Read in a, b, c and use the quadratic formula. If the discriminate b2-4ac is negative, display a message stating that there are no real solutions**.**

public class QuadraticEquation {

int a, b, c;

double root1, root2, d;

Scanner s = new Scanner(System.in);

void input()

{

System.out.println("Quadratic equation is in the form : ax^2 + bx + c");

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

a = s.nextInt();

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

b = s.nextInt();

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

c = s.nextInt();

}

void discriminant() {

d= (b\*b)-(4\*a\*c);

}

void calculateRoots() {

if(d>0)

{

System.out.println("Roots are real and unequal");

root1 = ( - b + Math.sqrt(d))/(2\*a);

root2 = (-b - Math.sqrt(d))/(2\*a);

System.out.println("First root is:"+root1);

System.out.println("Second root is:"+root2);

}

else if(d == 0)

{

System.out.println("Roots are real and equal");

root1 = (-b+Math.sqrt(d))/(2\*a);

System.out.println("Root:"+root1);

}

else

{

System.out.println("No real solutions. Roots are imaginary");

}

}

}

public class TestQE {

public static void main(String[] args) {

QuadraticEquation qe = new QuadraticEquation();

qe.input();

qe.discriminant();

qe.calculateRoots();

}

}

2). a) Design, Develop and Implement a Java program to add two given matrices using multidimensional arrays.

import java.util.Scanner;

class Matrix

{

int m, n, p, q, sum = 0, i,j,k;

Scanner in = new Scanner(System.in);

int First[][] = new int[10][10];

int Second[][] = new int[10][10];

int Result[][] = new int[10][10];

void input()

{

System.out.println("Enter the number of rows and columns of First matrix");

m = in.nextInt();

n = in.nextInt();

System.out.println("Enter elements of First matrix");

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

{

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

First[i][j] = in.nextInt();

}

System.out.println("Enter the number of rows and columns of Second matrix");

p = in.nextInt();

q = in.nextInt();

if (n != p)

{

System.out.println("The matrices can't be multiplied with each other.");

System.exit(0);

}

else

{

System.out.println("Enter the number of rows and columns of Second matrix");

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

{

for (j = 0; j <q; j++)

Second[i][j] = in.nextInt();

}

}

}

void add()

{

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

{

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

{

for(k=0;k<p;k++)

Result[i][j]=First[i][k]+Second[k][j];

}

}

}

void display()

{

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

{

for (j = 0; j <q; j++)

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

System.out.println();

}

}

}

public class Demo

{

public static void main(String args[])

{

Matrix MM=new Matrix();

MM.input();

MM.add();

MM.display();

}

}

2) b) Design, Develop and Implement a Java program to add and subtract two complex numbers and using the concept of constructor overloading.

public class ComplexNumber {

double real;

double imag;

ComplexNumber()

{

real = 0.0;

imag = 0.0;

}

ComplexNumber(double a)

{

real = a;

imag = 0;

}

ComplexNumber(double a, double b)

{

real = a;

imag = b;

}

ComplexNumber(ComplexNumber ob)

{

real = ob.real;

imag = ob.imag;

}

void add(ComplexNumber c1, ComplexNumber c2)

{

double realSum = c1.real+c2.real;

double imagSum = c1.imag+c2.imag;

System.out.println("Sum is "+realSum+"+i"+imagSum);

}

void sub(ComplexNumber c1, ComplexNumber c2)

{

double realDiff = c1.real-c2.real;

double imagDiff = c1.imag-c2.imag;

System.out.println("Difference is "+realDiff+"-i"+imagDiff);

}

}

public class ComplexNumberRun {

public static void main(String args[])

{

ComplexNumber cn1 = new ComplexNumber();

ComplexNumber cn2 = new ComplexNumber(10);

ComplexNumber cn3 = new ComplexNumber(20,30);

ComplexNumber cn4 = new ComplexNumber(cn3);

cn1.add(cn1,cn4);

cn2.sub(cn3,cn2);

}

}

3) a) Design, Develop and Implement a Java program to sort a list of elements in ascending and descending order and show exception handling.

**public** **class** Sort {

**void** sortAscend(**int** arr[], **int** n)

{

//int n = arr.length;

**for** (**int** i = 0; i < n-1; i++)

**for** (**int** j = 0; j < n-i-1; j++)

**if** (arr[j] > arr[j+1])

{

**int** temp = arr[j];

arr[j] = arr[j+1];

arr[j+1] = temp;

}

}

**void** sortDescend(**int** arr[], **int** n)

{

//int n = arr.length;

**for** (**int** i = 0; i < n-1; i++)

**for** (**int** j = 0; j < n-i-1; j++)

**if** (arr[j] < arr[j+1])

{

**int** temp = arr[j];

arr[j] = arr[j+1];

arr[j+1] = temp;

}

}

**void** printArray(**int** arr[], **int** n)

{

//int n = arr.length;

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

System.***out***.print(arr[i] + " ");

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

}

}

**public** **class** TestSort {

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

**int** n;

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

Sort s = **new** Sort();

System.***out***.println("Input number of integers to sort");

n = sc.nextInt();

**int** array[] = **new** **int**[10];

System.***out***.println("Enter " + n + " integers");

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

{

**try** {

array[i] = sc.nextInt();

}

**catch**(ArrayIndexOutOfBoundsException e){

System.***out***.println("Index out of bound");

System.*exit*(0);

}

}

System.***out***.println("Entered Array is");

s.printArray(array, n);

s.sortAscend(array,n);

System.***out***.println("Ascending order: ");

s.printArray(array,n);

s.sortDescend(array, n);

System.***out***.println("Descending order: ");

s.printArray(array,n);

sc.close();

}

}

3) b) Design, Develop and Implement a Java program to demonstrate multilevel inheritance by using super for calling the super class constructors. (Use Box, BoxWeight and BoxShipment classes)

class Box {

private double width;

private double height;

private double depth;

// construct clone of an object

Box(Box ob) { // pass object to constructor

width = ob.width;

height = ob.height;

depth = ob.depth;

}

// constructor used when all dimensions specified

Box(double w, double h, double d) {

width = w;

height = h;

depth = d;

}

// constructor used when no dimensions specified

Box() {

width = -1; // use -1 to indicate

height = -1; // an uninitialized

depth = -1; // box

}

// constructor used when cube is created

Box(double len) {

width = height = depth = len;

}

// compute and return volume

double volume() {

return width \* height \* depth;

}

}

// Add weight.

class BoxWeight extends Box {

double weight; // weight of box

// construct clone of an object

BoxWeight(BoxWeight ob) { // pass object to constructor

super(ob);

weight = ob.weight;

}

// constructor when all parameters are specified

BoxWeight(double w, double h, double d, double m) {

super(w, h, d); // call superclass constructor

weight = m;

}

// default constructor

BoxWeight() {

super();

weight = -1;

}

// constructor used when cube is created

BoxWeight(double len, double m) {

super(len);

weight = m;

}

}

// Add shipping costs.

class Shipment extends BoxWeight {

double cost;

// construct clone of an object

Shipment(Shipment ob) { // pass object to constructor

super(ob);

cost = ob.cost;

}

// constructor when all parameters are specified

Shipment(double w, double h, double d,

double m, double c) {

super(w, h, d, m); // call superclass constructor

cost = c;

}

// default constructor

Shipment() {

super();

cost = -1;

}

// constructor used when cube is created

Shipment(double len, double m, double c) {

super(len, m);

cost = c;

}

}

class DemoShipment {

public static void main(String args[]) {

Shipment shipment1 = new Shipment(10, 20, 15, 10, 3.41);

Shipment shipment2 =new Shipment(2, 3, 4, 0.76, 1.28);

double vol;

vol = shipment1.volume();

System.out.println("Volume of shipment1 is " + vol);

System.out.println("Weight of shipment1 is "+ shipment1.weight);

System.out.println("Shipping cost: $" + shipment1.cost);

System.out.println();

vol = shipment2.volume();

System.out.println("Volume of shipment2 is " + vol);

System.out.println("Weight of shipment2 is "+ shipment2.weight);

System.out.println("Shipping cost: $" + shipment2.cost);

}

}

4) a) Design, Develop and Implement a Java program to calculate the interest amount based on the rate of interest defined for different banks using the concept of interface. Also calculate and display the maturity amount.

**public** **interface** FixedDeposit {

**double** getMAmount();

**void** calculateInterest();

**void** getDetails();

}

**public** **class** CanaraBank **implements** FixedDeposit{

Scanner s = **new** Scanner(System.***in***);

String name;

**double** principal;

**double** period;

**double** roi = 8.5;

**double** interestAmt;

**public** **void** getDetails()

{

System.***out***.println("Enter your name");

name = s.nextLine();

System.***out***.println("Enter the Principal amount");

principal = s.nextDouble();

System.***out***.println("Enter the period of deposit");

period = s.nextDouble();

}

**public** **void** calculateInterest() {

interestAmt = (principal\*period\*roi)/100;

}

**public** **double** getMAmount() {

**double** totalBalance;

totalBalance = principal+interestAmt;

**return** totalBalance;

}

}

**public** **class** SBI **implements** FixedDeposit{

Scanner s = **new** Scanner(System.***in***);

String name;

**double** principal;

**double** period;

**double** roi = 8.75;

**double** interestAmt;

**public** **void** getDetails()

{

System.***out***.println("Enter your name");

name = s.nextLine();

System.***out***.println("Enter the Principal amount");

principal = s.nextDouble();

System.***out***.println("Enter the period of deposit");

period = s.nextDouble();

}

**public** **void** calculateInterest() {

interestAmt = (principal\*period\*roi)/100;

}

**public** **double** getMAmount() {

**double** totalBalance;

totalBalance = principal+interestAmt;

**return** totalBalance;

}

}

**public** **class** TestBank {

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

**double** mAmount;

SBI s = **new** SBI();

CanaraBank cb = **new** CanaraBank();

s.getDetails();

s.calculateInterest();

mAmount = s.getMAmount();

System.***out***.println("Dear "+s.name+" your Maturity Amount in SBI Bank is "+mAmount);

cb.getDetails();

cb.calculateInterest();

mAmount = cb.getMAmount();

System.***out***.println("Dear "+s.name+" your Maturity Amount in Canara Bank is "+mAmount);

}

}

4) b) Design, Develop and Implement a Java program to compute the surface area and volume of cylinder, cone and sphere. Create an abstract class “Solid” and the classes cylinder, cone and sphere have to inherit the common properties form the class “Solid”.

**public** **abstract** **class** Solid {

**double** r, h;

**abstract** **void** surfaceArea();

**abstract** **void** volume();

**void** readRadius()

{

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

System.***out***.println("Enter the radius");

r=sc.nextDouble();

}

**void** readHeight()

{

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

System.***out***.println("Enter the height");

h=sc.nextDouble();

}

}

**public** **class** Cone **extends** Solid{

**void** surfaceArea()

{

**double** area = (3.14 \* r)\*(r \* Math.*sqrt*(r\*r + h\*h));

System.***out***.println("Surface area of cone is "+area);

}

**void** volume()

{

**double** volume = 3.14 \* r \* r \* (h/3);

System.***out***.println("Volume of cone is "+volume);

}

}

**public** **class** Cylinder **extends** Solid {

**void** surfaceArea()

{

//System.out.println(r+" "+h);

**double** area = 3.14 \* r \* r \* h;

System.***out***.println("Surface area of cylinder is " +area);

}

**void** volume()

{

**double** volume = (2 \* 3.14 \* r \* h) + (2 \* 3.14 \* r \* r);

System.***out***.println("Volume of cylinder is "+volume);

}

}

**public** **class** Sphere **extends** Solid {

**void** surfaceArea()

{

**double** area = 4 \* 3.14 \* r \* r;

System.***out***.println("Surface area of sphere is "+area);

}

**void** volume()

{

**double** volume = 4.0/3 \* 3.14 \* r \* r \* r;

System.***out***.println("Volume of sphere is "+volume);

}

}

**public** **class** MySolid {

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

Solid s=**new** Cylinder();

s.readRadius();

s.readHeight();

s.surfaceArea();

s.volume();

s=**new** Cone();

s.readRadius();

s.readHeight();

s.surfaceArea();

s.volume();

s=**new** Sphere();

s.readRadius();

s.surfaceArea();

s.volume();

}

}

5) a) Design, Develop and Implement a Java program that creates three threads. First thread displays “Good Morning “every one second, the second thread displays “Hello” every two seconds and the third thread displays “Welcome” every three seconds.

public class One extends Thread {

public void run()

{

try

{

int i=0;

while (i<5)

{

sleep(1000);

System.out.println("Good morning");

i++;

}

}

catch(InterruptedException e)

{

System.out.println("Interrupted");

}

}

}

public class Two extends Thread {

public void run()

{

try

{

int i=0;

while (i<5)

{

sleep(2000);

System.out.println("Hello");

i++;

}

}

catch(InterruptedException e)

{

System.out.println("Interrupted");

}

}

}

public class Three extends Thread {

public void run()

{

try

{

int i=0;

while (i<5)

{

sleep(3000);

System.out.println("Welcome");

i++;

}

}

catch(InterruptedException e)

{

System.out.println("Interrupted");

}

}

}

}

}

public class MyThread {

public static void main(String[] args) {

One t1=new One();

Two t2=new Two();

Three t3=new Three();

t1.start();

t2.start();

t3.start();

}

}

5) b) Design, Develop and Implement a Java program to read from console and write these data into file, again read that file and print the content onto console.

public class IO

{

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

{

char c;

int i;

FileInputStream fin = new FileInputStream("Test1.txt");

FileOutputStream fout = new FileOutputStream("Test1.txt");

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

System.out.println("Enter characters, 'q' to quit.");

do{

c = (char) br.read();

if(c != 'q') fout.write(c);

} while(c != 'q');

System.out.println("Contents of the file are");

do{

i = fin.read();

if(i != -1) System.out.print((char) i);

} while(i != -1);

fin.close();

fout.close();

}

}

6) a) Design, Develop and Implement a java program to implement a stack using generic class and methods.

public class Stack<E> {

E stck[];

int top;

final int SIZE = 10;

@SuppressWarnings("unchecked")

Stack()

{

stck = (E[]) new Object [SIZE];

top = -1;

}

void push(E item)

{

if (top == SIZE-1)

System.out.println("Stack is full");

else

stck[++top] = item;

}

E pop()

{

if(top < 0) {

System.out.println("Stack underflow");

return null;

}

else

return stck[top--];

}

}

import java.util.Scanner;

public class TestStack {

public static void main(String[] args) {

Stack<Integer> mystack1 = new Stack<Integer>();

Stack<Double> mystack2 = new Stack<Double>();

Scanner s = new Scanner(System.in);

System.out.println("Enter elements into the Integer stack");

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

{

int n = s.nextInt();

mystack1.push(n);

}

System.out.println("Enter elements into the Double stack");

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

{

double m = s.nextDouble();

mystack2.push(m);

}

System.out.println("Elements of stack 1 ");

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

System.out.println(mystack1.pop());

System.out.println("Elements of stack 2 ");

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

System.out.println(mystack2.pop());

s.close();

}

}

6) b) Design, Develop and Implement a Java program to read a string from the keyboard using an appropriate UI and do the following

i. Extract the middle character of the string

ii. Check whether the string entered is palindrome or not

iii. Counting the number of vowels in the string

iv. Counting the total number of characters in this string**.**

package swings;

import javax.swing.\*;

import java.awt.event.\*;

public class StringOperations implements ActionListener {

JTextField tf1,tf2,tf3;

JButton b1,b2,b3,b4;

JLabel l1,l2;

StringOperations(){

JFrame f= new JFrame();

tf1=new JTextField();

tf1.setBounds(80,80,150,20);

tf2=new JTextField();

tf2.setBounds(80,130,200,20);

tf2.setEditable(false);

l1=new JLabel("Enter text");

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

l2=new JLabel("Result");

l2.setBounds(50,100,100,30);

b1=new JButton("Find Mid Character");

b1.setBounds(50,200,250,50);

b2=new JButton("Check Palindrome");

b2.setBounds(320,200,250,50);

b3=new JButton("Count Vowels");

b3.setBounds(50,300,250,50);

b4=new JButton("Find length");

b4.setBounds(320,300,250,50);

b1.addActionListener(this);

b2.addActionListener(this);

b3.addActionListener(this);

b4.addActionListener(this);

f.add(tf1);

f.add(tf2);

f.add(l1);

f.add(l2);

f.add(b1);

f.add(b2);

f.add(b3);

f.add(b4);

f.setSize(650,450);

f.setLayout(null);

f.setVisible(true);

}

public void actionPerformed(ActionEvent e) {

String s=tf1.getText();

int length;

length = s.length();

if(e.getSource()==b1) {

int mid;

mid = length/2;

if(length%2 != 0)

tf2.setText("The mid character is "+s.charAt(mid));

else

tf2.setText("The middle characters are "+s.charAt(mid-1)+""+s.charAt(mid));

}

else if(e.getSource()==b2) {

String reverse = "";

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

reverse = reverse + s.charAt(i);

}

if (s.equals(reverse))

tf2.setText("Entered string is a palindrome.");

else

tf2.setText("Entered string is not a palindrome.");

}

else if(e.getSource()==b3) {

int count = 0;

char ch;

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

{

ch = s.charAt(i);

if (ch == 'a' || ch == 'A' || ch == 'e' || ch == 'E' || ch == 'i' ||

ch == 'I' || ch == 'o' || ch == 'O' || ch == 'u' || ch == 'U')

count ++;

}

tf2.setText("The number of vowels: "+count);

}

else if(e.getSource()==b4) {

int i=0;

while (true)

{

try

{

s.charAt(i);

i++;

}

catch(StringIndexOutOfBoundsException ex)

{

tf2.setText("The length of string is "+i);

break;

}

}

}

}

public static void main(String[] args) {

new StringOperations();

}

}