**JAVA LAB PART A - ANSWERS**

**1.a. Write a java program to find the details of the students eligible to enroll for the**

**examination(Students & Department give the eligibility criteria for the enrollment class)**

**using interfaces.**

//attendence variable store no of class attended out of 54

interface student{

String getValue();

}

interface department{

float getattendance();

}

interface exam extends student,department{

float calattendance();

boolean eligible();

}

class check implements exam{

int sno;

String sname;

String clas;

float attendance;

float perc;

public check(int s, String name, String c, float a) {

sno=s;

sname=name;

clas=c;

attendance=a;

}

@Override

public String getValue() {

return ("\nStudent Number: " +sno +"\nStudent Name: " +sname

+"\nClass: " +clas +"\nAttendace: " +attendance + "\nAttendace Percentage: "+perc );

}

@Override

public float getattendance() {

return perc;

}

@Override

public float calattendance() {

perc = (attendance/54)\*100;

return perc;

}

@Override

public boolean eligible() {

if (perc>=85)

{return true;}

else

{return false;}

}

}

public class prog {

public static void main(String [] args) {

check s1 = new check(1,"A","5B",23);

s1.calattendance();

if(s1.eligible())

{

System.out.print("Eligible!");

System.out.println(s1.getValue());

}

else

{

System.out.print("Not eligible!");

System.out.println(s1.getValue());

}

}

}

**2.a. Write a java program to maintain the student details like USN, Dept names, 3 subject grades and SGPA in student package and keep the staff details such as Staffid, StaffName, designation and subjects handled in a staff package. In main class use these two packages details for Staff and Student classes and display the student and staff information as requested by the user.**

//student package

package Student;

public class stud\_details

{

String USN, Dept\_name, g1,g2,g3;

double sgpa;

public stud\_details(String u,String d,String gr1,String gr2,String gr3,double sg)

{

USN=u;

Dept\_name=d;

g1=gr1;

g2=gr2;

g3=gr3;

sgpa=sg;

}

public void display()

{

System.out.println(" USN : "+ USN+"\n Department name :"+ Dept\_name + "\n Grade 1 "+g1+

"\n Grade 2 "+g2+"\n Grade 3 "+g3+"\n SGPA: "+sgpa);

}

}

//staff package

package Staff;

public class staff\_details {

String Staffid, StaffName, Designation, subjects;

public staff\_details(String i,String n,String d,String s)

{

Staffid=i;

StaffName=n;

Designation=d;

subjects=s;

}

public void display()

{

System.out.println("\n Staffid: "+Staffid+"\n StaffName "+StaffName+"\n Designation "+ Designation +"\n Subject handled "+subjects);

}

}

//main

import Student.\*;

import Staff.\*;

public class prog

{

public static void main(String []args)

{

stud\_details s1 = new stud\_details("1MS15CS148","CSE","S","A","S",9.70);

s1.display();

staff\_details s2 = new staff\_details("TE001","ABC","HOD", "Mathematics");

s2.display();

}

}

**3.a. Write a java program to throw a exception for an employee details. If an employee name is a number, a name exception must be thrown. If an employee age is greater than 50, an age exception must be thrown Or else an object must be created for the entered employee details**

import java.util.Scanner;

class NameException extends Exception

{

public String toString()

{

return ("Name Exception");

}

}

class AgeException extends Exception

{

public String toString()

{

return ("Age Exception");

}

}

class Emp\_Details

{

int age;

String name;

Emp\_Details(int a,String n)

{

age=a;

name=n;

}

}

public class prog {

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

System.out.println("Enter your age: ");

int age = sc.nextInt();

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

String name = sc.next();

int nx=1, ax=1;

try

{

try

{

int ns = Integer.parseInt(name);

}

catch(Exception e)

{

nx=0;

}

if(nx==1)

throw new NameException();

}

catch(NameException e)

{

System.out.println(e);

ax=0;

}

try

{

if(age>50)

{

throw new AgeException();

}

}

catch(AgeException e)

{

System.out.println(e);

ax=0;

}

catch(Exception e)

{

System.out.println(e);

ax=0;

}

if(ax==1)

{

new Emp\_Details(age,name);

System.out.println("Object created successfully! \n Name: "+ name+" \n Age: "+ age);

}

}

}

**4.a. Write Java program to create an applet with text box. We must type a string in text box first. Then if we press “P” key check the given string is PALINDROME or not and the result must be displayed on the status bar.**

import java.applet.Applet;

import java.awt.\*;

import java.awt.event.\*;

public class prog extends Applet implements KeyListener

{

TextField t = new TextField();

Label l = new Label();

Label enter = new Label("Enter text:");

public void init()

{

setLayout(new GridLayout(3,1));

add(enter);

add(t);

add(l);

t.addKeyListener(this);

setVisible(true);

}

boolean palin(String s)

{

for (int i=0, j=s.length()-1; i < j; i++, j--)

if (s.charAt(i) != s.charAt(j))

return false;

return true;

}

@Override

public void keyPressed(KeyEvent e)

{

if (e.getKeyChar() =='p')

{

if (palin(t.getText()))

{

l.setText("Palindrome");

showStatus("Palindrome");

}

else

{

l.setText("Not a palindrome");

showStatus("Not a palindrome");

}

}

}

@Override

public void keyReleased(KeyEvent e) {

}

@Override

public void keyTyped(KeyEvent e) {

}

}

**5.a. Write a multithreaded program to create 2 threads. One thread should infinitely display the message as “welcome” while other thread should infinitely display the message “Goodbye”. Synchronize the execution of these threads. So that the messages are displayed alternately.**

class display

{

boolean value=false;

synchronized public void welcome()

{

while(!value)

{

try

{

wait();

}

catch(Exception e)

{

System.out.println(e);

}

}

System.out.println("Welcome");

value = false;

notify();

}

synchronized public void goodbye()

{

while(value)

{

try

{

wait();

}

catch(Exception e)

{

System.out.println(e);

}

}

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

value = true;

notify();

}

}

class t1 extends Thread

{

display p;

t1(display m)

{

p=m;

this.start();

}

public void run()

{

while(true)

{

p.welcome();

}

}

}

class t2 extends Thread

{

display p;

t2(display m)

{

p=m;

this.start();

}

public void run()

{

while(true)

{

p.goodbye();

}

}

}

public class prog

{

public static void main (String [] args)

{

display p = new display();

new t1(p);

new t2(p);

}

}

**6.a. Create a main class to prompt the user to enter his/her age and his CGPA. The user application for a job will be rejected either if his age is greater than 25 years or his CGPA is less than 8. You should declare two nested try-throw-catch blocks; one to handle the AgeOutOfRangeException and the other to handle the LowCgpaException. If the user enters acceptable age and CGPA, display the message “Your application is accepted and is under study”.**

import java.util.Scanner;

class LowCgpaException extends Exception

{

public String toString()

{

return "Low CGPA Exception raised!";

}

}

class AgeOutOfRangeException extends Exception

{

public String toString()

{

return "Age Out Of Range Exception raised!";

}

}

public class prog

{

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

System.out.println("Enter your Age:");

int age = sc.nextInt();

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

double cgpa = sc.nextDouble();

int k =1;

try

{

if(age<=25)

{

try

{

if(cgpa<8)

{

throw new LowCgpaException();

}

else

k=0;

}

catch (LowCgpaException e)

{

System.out.println(e);

}

}

else

{

throw new AgeOutOfRangeException();

}

}

catch(AgeOutOfRangeException e)

{

System.out.println(e);

}

if(k==0)

System.out.println("Your application is accepted and is under study!");

}

}

**7.a. Write a Java program to create a super class Record has been defined to store the names and ranks of 50 students. Define a sub class Rank to find the highest rank along with the name. The details of both classes are given below**

**Class Name :Record**

**Data Members : name[ ], rnk[ ] (store names and respective ranks in an array**

**Member functions: Record() : constructor to initialize data members**

**void readvalues() : to store names and ranks**

**void display() : displays the names and the corresponding ranks**

**Class name : Rank**

**Data Members : index(integer to store the index of the topmost rank)**

**Member functions :**

**Rank() : constructor to invoke the base class constructor and to initialize index to 0.**

**void highest() : finds the index location of the topmost rank and stores it in index without sorting the array**

**void display() : displays the name and ranks along with the name having the topmost rank.**

import java.util.Scanner;

class Record

{

String[] name= new String[50];

int[] rnk= new int[50];

Record()

{

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

}

void readvalues()

{

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

{

Scanner sc = new Scanner(System.in);

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

name[i]=sc.next();

System.out.println("Enter your rank: ");

rnk[i]=sc.nextInt();

}

}

void display()

{

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

{

System.out.println("\n Name: " + name[i] + "\n Rank: " + rnk[i]);

}

}

}

class Rank extends Record

{

int index;

Rank()

{

super();

index=0;

}

void highest()

{

int max=999;

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

{

if(max>=rnk[i])

{

index =i;

max=rnk[i];

}

}

}

void display()

{

super.display();

System.out.println("\n\n Top most rank is : " +name[index] + "\n Rank is " + rnk[index] );

}

}

public class prog

{

public static void main(String args[])

{

Rank r = new Rank();

r.readvalues();

r.highest();

r.display();

}

}

**8.a. Write a java program to implement the following :**

**Consider a restaurant that has one chef and one waitperson. The waitperson must wait for the chef to prepare a meal. When the chef has a meal ready, the chef notifies the waitperson, who then gets the meal and goes back to waiting. The chef represents the producer, and the waitperson represents the consumer.**

class wait\_eat

{

int n;

boolean value=false;

synchronized public int eat()

{

while(!value)

{

try

{

wait();

}

catch(Exception e)

{

System.out.println(e);

}

}

System.out.println("Got Meal number: "+n);

value = false;

notify();

return n;

}

synchronized public void wait1(int n)

{

while(value)

{

try

{

wait();

}

catch(Exception e)

{

System.out.println(e);

}

}

this.n= n;

System.out.println("Put Meal number: "+n);

value = true;

notify();

}

}

class t1 extends Thread

{

wait\_eat p;

t1(wait\_eat m)

{

p=m;

this.start();

}

public void run()

{

int i=0;

while(true)

{

p.wait1(i++);

try {

sleep(5000);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

}

}

class t2 extends Thread

{

wait\_eat p;

t2(wait\_eat m)

{

p=m;

this.start();

}

public void run()

{

while(true)

{

p.eat();

try {

sleep(5000);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

}

}

public class prog

{

public static void main(String args[])

{

wait\_eat p = new wait\_eat();

new t1(p);

new t2(p);

}

}

**9.a. Define a class called Library with the following description:**

**Instance variables/data members:**

**Int acc\_num – stores the accession number of the book**

**String title – stores the title of the book stores the name of the author**

**Member Methods:**

**(i) void input() – To input and store the accession number, title and author.**

**(ii)void compute – To accept the number of days late, calculate and display and fine charged**

**at the rate of Rs.2 per day.**

**(iii) void display() To display the details in the following format:**

**Accession Number Title Author**

**Write a main method to create an object of the class and call the above member methods.**

import java.util.Scanner;

public class Library

{

int acc\_num;

String title, author;

Scanner sc = new Scanner(System.in);

void input()

{

System.out.println("Enter Accession Number:");

acc\_num = sc.nextInt();

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

title = sc.next();

System.out.println("Enter Author Name:");

author = sc.next();

}

void compute()

{

System.out.println("Enter Number of days late:");

int late = sc.nextInt();

int fine = late \* 2;

System.out.println("The fine is: " + fine + " rupees");

}

void display()

{

System.out.println(acc\_num + " " + title + " " + author + " ");

}

}

//main method

public class prog

{

public static void main(String args[])

{

Library l = new Library();

l.input();

l.compute();

l.display();

}

}

**10.a. Write Java program to create a package called AdvMath. which has two classes. In main class use this package display the result as requested by the user.**

**(i) to calculate y = sinx+cosx+tanx**

**(ii) to print Pythagorean triplets**

//trig

package AdvMath;

public class Trig

{

double x,total;

public void cal(int m)

{

x = Math.toRadians(m);

total = Math.sin(x)+ Math.cos(x)+ Math.tan(x);

}

public void display()

{

System.out.println("sinx + cosx + tanx = " + total);

}

}

//pythagorean triplets

package AdvMath;

public class Pythagorean

{

public void triplets(int limit)

{

// triplet: a^2 + b^2 = c^2

int a, b, c=0;

// loop from 2 to max\_limitit

int m = 2;

// Limiting c would limit all a, b and c

while (c < limit)

{

// now loop on j from 1 to i-1

for (int n = 1; n < m; ++n)

{

// Evaluate and print triplets using

// the relation between a, b and c

a = m\*m - n\*n;

b = 2\*m\*n;

c = m\*m + n\*n;

if (c > limit)

break;

System.out.println(a + " " + b + " " +c);

}

m++;

}

}

}

//main

import AdvMath.\*;

public class prog

{

public static void main(String args[])

{

Trig t=new Trig();

t.cal(45.0);

t.display();

Pythagorean pt = new Pythagorean();

pt.triplets(15);

}

}

**11.a. Write a java program to accept a string. Convert the string to uppercase. Count and output the number of double letter sequences that exist in the string.**

**Sample Input: “SHE WAS FEEDING THE LITTLE RABBIT WITH AN APPLE”**

**Sample Output: 4**

import java.util.Scanner;

public class prog

{

public static void main(String args[])

{

String s = "";

int count = 0;

System.out.println("Enter your string:");

Scanner sc = new Scanner(System.in);

s += sc.nextLine();

s=s.toUpperCase();

//System.out.println(s);

s=s.replaceAll(" ", "");

//System.out.println(s);

for(int i=0;i<s.length()-1;i++)

if(s.charAt(i) == s.charAt(i+1))

count+=1;

System.out.println(count);

}

}

**12.a. Write a java program to create five threads with different priorities. Send two threads of highest priority to sleep state. Check the aliveness of the threads.**

class mythread extends Thread

{

int n;

mythread(int n)

{

this.n=n;

}

public void run()

{

System.out.println("Running thread "+n);

while(true) {}

}

}

public class prog

{

public static void main(String []args)

{

mythread t1 = new mythread(1);

mythread t2 = new mythread(2);

mythread t3 = new mythread(3);

mythread t4 = new mythread(4);

mythread t5 = new mythread(5);

t1.setPriority(9);

t2.setPriority(8);

t3.setPriority(7);

t4.setPriority(6);

t5.setPriority(5);

t1.start();

t2.start();

t3.start();

t4.start();

t5.start();

try

{

t1.sleep(100);

t2.sleep(100);

}

catch (InterruptedException e)

{

e.printStackTrace();

}

if(t1.isAlive())

System.out.println("Active t1");

else

System.out.println("Sleep t1");

if(t2.isAlive())

System.out.println("Active t2");

else

System.out.println("Sleep t2");

if(t3.isAlive())

System.out.println("Active t3");

else

System.out.println("Sleep t3");

if(t4.isAlive())

System.out.println("Active t4");

else

System.out.println("Sleep t4");

if(t5.isAlive())

System.out.println("Active t5");

else

System.out.println("Sleep t5");

}

}

**13.a. Write a Java Program to Implement an abstract class Reservation and two classes ReserveTrain and ReserveBus**

**Define a Reservation abstract class with following characteristics**

**Method reserve which takes integer value seats and typeOfSeat as parameters and returns boolean type.**

**Method getAvailableSeats which return a number of seat remaining.**

import java.util.Scanner;

abstract class Reservation{

abstract public boolean reserve(int seat,String typeOfSeat);

abstract public int getAvailSeats();

}

class ReserveTrain extends Reservation{

int seat;

static int nost=72;

String tos;

public boolean reserve(int seat, String typeOfSeat) {

this.seat=seat;

tos=typeOfSeat;

if(this.seat>0 && tos!=null && this.seat<=nost) {

System.out.println("seats available "+getAvailSeats());

return true;}

else

return false;

}

public int getAvailSeats() {

// TODO Auto-generated method stub

nost = nost-seat;

return (nost); //available minus requested for reservation

}

}

class ReserveBus extends Reservation{

int seat;

static int nosb=50;

String tos;

public boolean reserve(int seat, String typeOfSeat) {

this.seat=seat;

tos=typeOfSeat;

if(this.seat>0 && tos!=null &&this.seat<=nosb) {

System.out.println("seats available "+getAvailSeats());

return true;}

else

return false;

}

public int getAvailSeats() {

nost = nost-seat;

return (nost); }

}

public class Q13A {

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

ReserveTrain rt=new ReserveTrain();

ReserveBus rb=new ReserveBus();

int op,seat;

String tos;

boolean b;

while(true) {

System.out.println("Select an option 1.Reserve Train

2.Reserve Bus");

op=sc.nextInt();

switch(op) {

case 1:

System.out.println("number of seats availabe" +rt.nost);

System.out.println("enter type of seat (ac/sleeper/general)");

tos=sc.next();

System.out.println("enter number of seats to reserve ");

seat=sc.nextInt();

b=rt.reserve(seat, tos);

System.out.println("reservation successful? "+b);

break;

case 2:

System.out.println("number of seats availabe" +rb.nosb);

System.out.println("enter type of seat (ac/sleeper/general)");

tos=sc.next();

System.out.println("enter number of seats to reserve ");

seat=sc.nextInt();

b=rb.reserve(seat, tos);

System.out.println("reservation successful? "+b);

break;

}

}

}

}

**14.a. Write a Java program to create an applet when we drag the mouse, the path of the mouse pointer must be drawn as a rectangle.**

import java.applet.Applet;

import java.awt.Graphics;

import java.awt.event.MouseEvent;

import java.awt.event.MouseListener;

import java.awt.event.MouseMotionListener;

public class prog extends Applet implements MouseMotionListener,MouseListener

{

int startx, starty, x, y, height, width;

public void init()

{

addMouseListener(this);

addMouseMotionListener(this);

}

@Override

public void mouseDragged(MouseEvent e) {

x=startx;

y=starty;

height=e.getX()-startx;

width=e.getY()-starty;

if(height<0)

{

x=e.getX();

height=Math.abs(height);

}

if(width<0)

{

y=e.getY();

width=Math.abs(width);

}

repaint();

}

@Override

public void mouseReleased(MouseEvent e)

{

startx=e.getX();

starty=e.getY();

showStatus("mouse pressed");

}

public void paint(Graphics g)

{

g.drawRect(x,y,height,width);

}

@Override

public void mouseMoved(MouseEvent e) {}

@Override

public void mouseClicked(MouseEvent e) {}

@Override

public void mouseEntered(MouseEvent e) {}

@Override

public void mouseExited(MouseEvent e) {}

@Override

public void mousePressed(MouseEvent e) {}

}