LAB - 1

Date :

1 Parseint ( )

- to find Area of rectangle

```
class RectangleArea {
public static void main (String args [] ) {
int length , breath;
length = Integer . parseint (args [0]);
breath = integer. parseint (args [1]);
int area = length * breath;
System . out. println (" length of rectangle = " + length);
System . out . println ("breath of rectangle=" + breath);
System . out . println (" area of rectangle=" + area);
}}.
```

2 Scanner

```
Import java.util. Scanner;
class Hello world {
public static void main (string args [])
{
int a ; float b ; string s;
Scanner in = new Scanner (System.in);
System. out . println (" Enter a string);
s= in.nextline ();
System.out.println (" entered string" + s);
System . out. println (" Enter an Integer");
a= in. nextint ();
System.out. println (" You entered an Integer" + a);
System. out. println ("Enter a float");
b= In. nextfloat ();
System . out. println ("Entered float" + b);
}
}
```

3 factorial

```
class factorial {
public static void main(String args[])
{
 int fac=1;
 System.out.println("Enter a number");
 Scanner sc = new Scanner (Scanner.in);
 int n = sc.nextInt();
 for (int i=1 ; i<=n; i++){
    fac = fac * i;
 }
 System.out.println("the factorial:" + fac);
}
}
```

4 Palindrome

```
class Palindrome {
public static void main (String args[])
{
int n, t, rem, rev = 0;
Scanner sc = new Scanner (System.in);
System.out.println("Enter a 5 digit number:");
n=sc.nextIn();
t=n;
while (t >0) {
rem = t-1.10;
rev = rev* 10 + rem;
t = t/10
}
if (rev==n){
System.out.println("Palindrome);
```

Date :

```
}
else {
System.out.println("not Palindrome");
}
}
}
```

5 WAP to find the sum of digits in a 5 digit number.

```
Class sumofdigits() {
public static void main (String args[]) {
  long number, sum;
Scanner Sc = new Scanner (System.in);
System.out.print ("Enter a 5-digit number:");
number = Sc.next long ();
for (sum=0; number != 0; number= number /10) {
    sum= sum + number % 10;
}
System.out.println ("Sum of digits ;" + sum);
  }
}
```

6 Array

```
Class AutoArray {
public static void main (String args[]) {
int month_days[] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
System.out.println ("April has" + month_days[3] + "days.")
}
}
```

Date :

× Quadratic Equation

```
Import java.util.Scanner;
class Quadratic
{
 int a,b,c
 double r1,r2,d;
 void getd ()
{
    Scanner s = new Scanner (System.in);
    System.out.println ("Enter the coefficients of a,b,c");
    a= s.nextint();
    b= s.nextint();
    c=s.nextint();
 }
 void compute ()
 {
  while (a==0)
  {
    System.out.println ("Not a quadratic equation");
    system.out.println ("Enter a non zero value for a!");
    Scanner s= new scanner (System.in);
    a= s.nextint();
 }
   d=b*b-4*a*c;
   if (d==0)
{
   r1=(-b)/(2*a);
   System.out.println ("Roots are real and equal");
   System.out.println ("Root1 = Root2= " + r1);
 }
 else if (d>0)
{
 r1= ((-b)+(Math.sqrt(d)))/(double)(2*a);
```

```
r2 = ((-b) -(Math.sqrt(d))) / (double) (2*a);
System.out.println("Roots are real and distinct");
System.out.println("Root1 = " + r1 + " Root2 = " + r2);
}

else if (d<0)
{
  System.out.println("Roots are imaginary");
  r1 = (-b) / (2*a);
  r2 = Math.sqrt(-d) / (2*a);
  System.out.println("Root1 = " + r1 + "+i " + r2);
  System.out.println("Root1 = " + r1 + "-i " + r2);
}
}
Class QuadraticMain
{
  public static void main(String args[])
  {
    Quadratic q = new Quadratic();
    q.getd();
    q.compute();
  }
}
```

9

9/1/2024

LAB-2

Date :

× Develop a java program to create a class student with members usn, name, an array credits and an import array marks. Include methods to accept & display details & a method to calculate SGPA

=)
```
import java.util Scanner;
class Student {
int subjectMarks;
int credits;
int grade;
}

Class Student {
String name;
String usn;
double SGPA;
Subject [] subjects;
Scanner S;

Student() {
int i;
subject = new subject [9];
for (i=0; i< 9; i++)
  subject[i] = new subject();
S = new Scanner (System.in);
}

void getStudentDetails () {
System out println ("Enter Student name");
name = S.next()
System.out println ("Enter student usn");
usn = S.next();
}

void getMarks () {
int i;
for(i=0; i<9; i++) {
System.out.println ("Enter marks for subject" + (i+1));
Subject [i] subjectMarks = S.nextInt();
System.out.println ("Enter credits for subject" + (i+1));
```

Date :

```
subject[i] credits = s.nextInt()
if (subject[i].subjectMarks >= 90) {
    subject[i] grade = 10;
}
else if (subject[i].subject Marks >= 90 && subject[i].marks >= 80)
{
    subject[i].grade = 9;
}
else if (subject[i].subject Marks >= 80 && subject[i].marks >= 70)
    subject[i].grade = 8;
}
else if (subject[i].subjectMarks >= 60 && subject[i].marks >= 60)
    subject[i].grade = 7;
}
else if (subject[i].subjectMarks >= 50 && subject[i].marks < 60)
    subject[i].grade = 6;
}
else if (subject[i].subjectmarks >= 40 && subject[i].marks < 50
{
    subject[i].grade = 5;
}
else {
    system out.println("failed");
    system.exit(0);
}
}
}

void computeSGPA () {
    int total Credits = 0;
    int credits Gained = 0;
    int i;
```

```
for (i=0; i<9; i++)
{
 total credits += subject [i] credits;
 credits gained += subject [i]. credits * subject[i]
}
SGPA = (double) credits Gained / total credits;
}
void display Result () {
 System. out. println("Name =" + name);
System. out. println ("USN = " + usn);
System. out. println ("SGPA =" + SGPA);
}
}
public class Main {
  public static void main (String args [] ) {
   Student s1 = new student ()
   s1. getStudent Details ();
   s1. get Marks ()
   s1. compute SGPA ();
   s1. display Result ();
   }
}
```

Output

```
Enter student name
Nischal
Enter USN
1BM22CS182
Enter marks for Subject 1
90
Enter credits for subject 1
4
Enter marks for subject 2
89
```

Date :

Enter credits for subject 2
4
Enter marks for subject 3
85
Enter credits for subject 3
4
Enter marks for subject 4
92
Enter credits for subject 4
3
Enter marks for subject 5
90
Enter credits for subject 5
3
Enter marks for subject 6
95
Enter credit for subject 6
2
Enter marks for subject 7
91
Enter ~~marks~~ credit for subject 7
1
Enter marks for subject 8
85
Enter credit for subject 8
1

9

Student Name : Nischal
Student USN : 1BM22CS182
SGPA : 9.590990.

9/1/24

Date :

Lab - 3

Create a class Book which contains four members, name, author, price, num-pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a toString() method that would display the complete details of the book. Develop a Java program to create n book objects.

=)
```
Import.java.util Scanner;

class Book {
    String name;
    string author;
    int price;
    int numpages;

Book(String name, string author, int price, int numpage)
{
    This.name = name;
    This.author = author;
    this.price = price;
    This.numpages = numpages;
}
public string to string() {
    string bookDetails = "Bookname" + this.name + "\n" +
                         "Author name" + this.author + "\n" +
                         "Price" + this.price + "\n" +
                         "Number of pages" + this.numpages + "\n";
    Return bookDetails;
}
}

public class Main {
public static void main (string [] args) {
    Scanner S = new Scanner (System.in);
    int n = S.nextInt();
    Book [] books = new Books [n];
```

Date :

```
for (int i = 0 ; i<n ; i++)
{
System.out.println("Enter details for book" + (i+1) + ":");
System.out.print ("Name ");
String name = S.next ();

System.out.print ("Author : ");
String author = s.next ();

System.out.print ("Price :");
int price = S.next int ();
book[i] = new Book(name, author, price, numpages); {
book.out.println("\n Details of the book :");
  for (int i=0 ; i<n ; i++) {
      System.out.println("\n Details of the book :");
      for (int i=0; i<n; i++) {
      System.out.println ("Book" + (i+1) + ":\n" + books[i].
                                                to string())
  }
}
}
```

Output

```
Enter the number of books : 2
Enter the details for book : 1
Name : Trump
Author : donald
Number of pages : 250
Enter details for book 2
Name : friends
Author : Dale
Price : 400
Number of pages 200
```

```
Details of books :
Book 1 :
Book name : Trump
Author name : donald
Price : 800
Number of pages : 250

Book 2
Book name : friends
Author name : Dale
Price : 400
Number of pages : 220
```

LAB - 9

Date : ?

Develop a Java program to create an abstract class named shape that contains 2 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.

=>
```
abstract class Area {
  int a, b;

  ~~Abstra~~ Abs Area (int x, int y) {
      a=x
      a=y;
  }


abstract void Area();
class Re extend Abs Area {
  rec(int a, int b) {
      super(a, b);
  }
  void Area() {
  System.out.println("the area of the rectangle is" + a*b);
  }
}
class tri extends Abs Area {
    tri(int a, int b) {
        Super (a, b);
    }
void area() {
System.out.println("the area of the triangle is" + (a*b)/2);
}
}
```

Date :

```
Class cir extends Abs Area {
    cir (int a) {
    Super (a);
  }
  void area () {
  System.out.println ("the area of circle is : "+3.14*a*a);
  }
}

Class Main {
public Static void main (string args [] ) {
Abs Area r = new rec (3,4);
r.area ();
Abs Area t = new tri (4,5);
t.area ();
Abs Area c = new cir (2);
c.area ();
  }
}
```

Output

The area of rectangle is : 12

The area of triangle is : 10

The area of the circle is : 12.56.

Pr
21/10/24

LAB-5

Date :

Develop a java program to create a class Bank that maintains two kind of account for its customers, one called savings account & the other current Account. The savings account provides compound interest & withdrawal facilities but no check book facility. The current account provides check book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed.

=>

```
import java.util.Scanner;
Class Account
{
String CustomerName;
double AccNo, Balance;
Account (String CustomerName, double AccNo, double Balance)
{
this.CustomerName = CustomerName;
this.AccNo = AccNo,
this.Balance = Balance;
}
Public void deposit (double Amount)
{
Balance + = Amount;
System.out.println ("Deposit Sucessful");
Display Balance ();
}
Void Display Balance ()
{
System.out.println ("BALANCE :", +Balance);
}
}
Class CurrentAccount extends Account
{
```

Date :

```
double MinBalance = 500.0;
double Charges = 10.0;
CurrentAccount (string CustomerName, double AccNo,
                double Balance)
{
Super ( CustomerName, AccNo, Balance);
}
void Withdraw (double Account)
{
if (Balance >= Amount)
{
Balance -= Amount;
DisplayBalance ();
}
else
{
System.out.println ("insufficient Balance");
}
}}
Class Bank
{
public static void main (string args[])
{
String CustomerName;
double AccNo, Balance; double amt, amt1;
Scanner in = new Scanner (system.in);
System.out.println ("enter name:");
CustomerName = in.next();

System.out.println ("enter AccNo");
AccNo = in.nextDouble();
System.out.println ("enter Balance:");
Balance = in.nextDouble ();
System.out.println ("enter amount to deposite");
```

Date :

```
amt= in.nextDouble();
System.out.println("enter amount to withdraw");
amt1 = in.nextDouble();

Kreate CurrentAccount c=new
CurrentAccount (customerName, AccNO, Balance);
C.Deposit (amt);
C.withdraw(amt1);
C.updateBalance ();
System.out.println(" ");
SavingsAccount s=new
SavingsAccount (customerName , AccNO, Balance);
S. Deposit (amt);
S. withdraw (amt1);
S.updateBalance();
}
}
```

Output

```
Enter Name :
Nischal
Enter Acc NO
174
Enter Balance
87000
Enter Amount to deposit
12000
Enter Amount to withdraw
9000

DEPOSIT DETAILS
BAIANCE : 99000
9000 withdrawn Sucessfully
```

Date :

BALANCE : 90000

DEPOSIT SUCESSFUL
BALANCE : 99000
BALANCE : 90000

Date :

LAB - 6

```
// Student
package CIE
public class student
{
  public string name;
  public string USN;
  public int sem;

public student(string name, string USN, int sem)
{
  this.name = name
  this.USN = USN
  this.sem = sem;
}
}

// Internal;
package CIE;

public class Internals extends ~~CIE~~ CIE student
{
  public int[] InternalMarks;

public Internals (string name, string USN, int sem, int[] internal marks)
{
  super (name, USN, sem);
  this.InternalMarks = InternalMarks;
}
}

// External
package SEE
import CIE student;
public class Externals extends student
```

Date :

```
{
 public int[] see Marks;
 public Externals(string name, String usn, int sem,
                                  int[] seemarks)
 {
    Super (Name, USN, Sem);
    this. See Marks = See Marks;
  }
}

// find marks

import CIE Student;
import CIE Internals;
import SEE. External;
import. java. uhl. scanner

public class find Marks
{
 public static void main (string [] Args)
  {
    Scanner sl = new scanner (system. in);
    System out. println ("Enter no. of students:");
    int n = sl. next Int();

 string [] names = new string [n];
 string [] USN = new string [n];
 int [] Sem = new int [n];
 int [] [] internal Marks = new int [n] [s]
 int [] [] See marks = new int [n] [s];

 for (int i = 0; i < n; i++).
 {
  System. out. println ("Enter details for student" + (i+1) +
  System. out. println ("Name");
```

Date :

```
names[i]= sl.next();
System.out.println("USN");
usn[i] = sl.next();
System.out.println("SEM");
sem[i] = sl.nextInt();
System.out.println("Enter Internal Marks for 5 courses");

for (int j=0; j<5; j++)
{
~~System.out.println("Name:")~~
System.out.println("course" +(j+1)+ ":");
InternalMarks[i][j] = sl.nextInt();
}
System.out.println("Enter internal marks for 5 courses:");
for (j=0; j<5; j++)
{
System.out.println("course" +(j+1)+ ":");
seeMarks[i][j] = sl.nextInt();
}
int[][] finalMarks = new int[n][5];
for (i=0; i<n; i++)
{
Internals Il =new Internals(names[i], usn[i], sem[i], I
Externals El=new Externals(names[i], usn[i], sem[i], se

for (int j=0; j<5; j++);
{
finalMarks[i][j] = Il.Internalmarks[i]+ El. seeMarks[j]
}
System.out.println("final Marks for" + n + " students in 5
courses

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

```
System.out.println(name[i] + ":");
for (int j=0 ; j<5; j++)
{
System.out.println(finalmarks[i][j] + ":");
}
    System.out.println();
}
Sl.close();
}
}
}
```

Output :

```
Enter number of students : 1
Enter name : A
Enter usn : 1
Enter sem : 1
Enter internal marks for 5 courses;
 20
 30
 10
 9
 9
Enter see marks for 5 courses
 10
 9
 9
 10
 9
FINAL MARKS of 5 COURSES
A : 124.
```

Date :

LAB 7

write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called "father" and derived class called "son" which extends base class. In father class implement a constructor which takes the age and throws the exception WrongAge() when the input age <0. In Son class, implement a constructor that calses both father and son's Age & throws an exception if sons age >= father Age.

=)

```
import Java.util Scanner;

class WrongAge extends Exception {
    WrongAge(string message){
      Super (message);
    }
}

Class Input Scanner {
    Scanner Sc = new Scanner (system.in)
}

class father extends Input Scanner {
    father() throws WrongAge {
    System.out.println("Enter father Age");
    fatherAge = Sc.next Int();
    if (fatherAge < 0) {
      throw new WrongAge ("Age cannot be registered");
    }
}
void display () {
  System.out.println("father age is " + fatherAge);
}
}
```

Date :

```
class Son extends father {
int SonAge;
Son () throws WrongAge {
System.out.println ("Enter Son's Age");
son Age = Sc.nextInt();
if (sonAge > fatherAge) {
throw new WrongAge ("Son's Age cannot be greater
                                than father's Age");
}
else if (sons Age < 0) {
throw new WrongAge ("Age cannot be registered
}
}
void display () {
System.out.println ("Son's Age is" + SonsAge)
}
}

public class Exception Handling {
public static void main (String [] args) {
try {
son son = new Son ();
son.display ();
}
catch (WrongAge e) {
System.out.println ("Exception" + e.getMessage(
}
}
}
```

Date :

Output

Enter fathers Age
50
Enter son's Age
21
fathers age is 50
Sons age is 21

Enter fathers Age
11
Enter son's Age
20
Error: father's Age must be greater than sons age

Enter fathers Age
-1
Error: Age cannot be negative.

30/1/2024

Date :

LAB- 8

* Write a program which creates two threads, one thread displaying "BMS College of Engineering" once every ten seconds & another displaying "CSE" once every two seconds

=) Class DisplayThread extends thread {
Private String message ;
private int interval ;
private boolean running = true ;
private (display Thread (String message, int interval)
{
This. message = message ;
This . message = interval ;
}
public void run ()
{
while (running)
{
System. out. println (message);
try {
Thread. slep (interval);
}
catch (Interrupted Exception e)
{
e. print StackTrace ();
}
}
}
public void stopthread ()
{
running = False
}
}
Class Main

Date :

```
{
public static void Main (string Args[])
DisplayThread bms = new DisplayThread("BMS", 10 sec)
DisplayThread CSE = new DisplayThread("CSE", 2 sec)
bms.start();
CSE.start();
bms.stopThread();
CSE.stopthread()
}
```

Output

BMS
CSE
CSE
CSE
CSE
BMS

late submission

23/8/24

Date :

LAB - 9

W.A.P that creates a user interface to perform integer division. The user enters two no.s in the text field, Num1 & Num2. The division of Num1 & Num2 is displayed in the Result field when the divide button is clicked. If Num1 & Num2 were not an integer, the program would throw an NumberFormat Exception. If Num2 were zero, the program would throw an Arithmetic Exception. Display the exception in a message display box

=)

```
import java.swing.*;
import java
import java

class swingDemo {
    SwingDemo() {
    Jframe jfrm = new Jframe("Divider App");
    Jfrm.setsize(275, 150);
    jfrm setLayout(new flowlayout());
    jfrm setDefaultCloseOperation(Jframe.EXIT_ON_CLOSE);

    Jlable jlab = new Jlable("Enter the divider & the dividend)
    JTextfield ajTF = new JTextfield(8);
    JTextfield bjTF = new JTextfield(8);

    JButton button = new JButton("calculate");

    Jlable err = new Jlable();
    Jlable alab = new Jlable();
    Jlable blab = new Jlabel()
    Jlabel ouslab = new Jlabel();
```

Date :

```
jfrm.add(err);
jfrm.add(jlab);
jfrm.add(ajtf);
jfrm.add(bjtf);
jfrm.add(button);
jfrm.add(alab);
jfrm.add(blab);
jfrm.add(anslab)

ActionListener = new ActionListener() {
public void actionPerformed(ActionEvent ect) {
System.out.println("Action event from a text field");
}
}
ajtf.addActionListener(l);
bjtf.addActionListener(l);
button.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent ect)
{
  try {
    int a = Integer.parseInt(ajtf.getText());
    int b = Integer.parseInt(bjtf.getText());
    int ans = a / b;

    alab.setText("\n A = " + a);
    blab.setText("\n B = " + b);
    anslab.setText("\n Ans = " + ans);
  }
  catch (NumberFormatException e) {
    alab.setText("");
    blab.setText("");
    anslab.setText("");
    err.setText("Enter Only Integer");
  }
```

Date :

```
catch (ArithmeticException e)
{
  alab.setText(" ");
  blab.setText(" ");
  anslab.setText(" ");
  err.setText("B should be NON ZERO");
}
}
}
jfrm.setVisible(true);
}

public static void main(String args[])
{
  SwingUtilities.invokeLater(new Runnable(){
    public void run(){
      new SwingDemo()
    }
  });
}
}
```

<u>Output</u>

Enter the divider & the dividend

20	4

Calculate

A = 20 B = 4 Ans = 5

Enter the divider & the divident.

10	0

B should be Non Zero.

Date :

AWT Functions

JFrame : It is a class in java that is part of the swing library which is used for creating graphical user interfaces in java operations.

Set Size ( ) : setsize() is a method which is used with components such as JFrame, JPanel, etc to set their size

Set Layout ( ) : It is a method which is used to set the layout manager for a container such as JFrame or any other container component.

JLabel : It is a class which is used to display non-editable text or image or GUI

Set default close operation : It is a method in java swing used to specify the default close operation for a 'JFrame'.

JTextfield : Class in java swing provides a text input field in a GUI It allows user to enter and edit single text line.

Add Action listener : It is a method in java swing that is used to register an action for a component eg. button that generates action events.

SetText : It is a method used in java swing to set the text content of text based component

LAB 10

Date :

Demonstrate Inter process communication & deadlock

```
Class Q
int n,
boolean value set = false;
synchronized int get() {
while (!value set)
try {
    System.out.println(":\n Consumer waiting \n");
wait();
} catch (InterruptedException e) {
    system.out.println("Interrupted Exception caught")
  }
System.out.println("Got :" + n);
value set = false,
System.out.println("\nProducer waiting \n");
wait();
} catch (InterruptedException e) {
System.out.println("Interrupted Exception caught ");
}
this.n = n;
value set = true
System.out.println("Put :"+n);
System.out.println("\n Intimate consumer \n");
notify();
}
}
Class Producer implements Runnable {
Q q;
producer (Q q) {
        this.q = {
            new thread (this producer") start()
        }
```

Date :

```
print void run () {
  int i = 0;
  while (i<15) {
    q.put(i++);
  }
}

class Consumer implements Runnable {
  Q q;
  Consumer (Q q) {
    this.q = q;
    new Thread(this, "Consumer").start();
  }
  public void run () {
    int i = 0;
    while (i<15) {
      int r = q.get();
      System.out.println("Consumed: " +r);
      i++;
    }
  }
}

class Main {
  public static void main (String args[]) {
    Q q = new Q();
    new producer(q);
    new Consumer(q);
    System.out.println("Press Ctrl C to stop");
  }
}
```

Date :

Output

```
Press control-c to stop
Put : 0
Intimate consumes
Producer waiting
Got 0
Intimate producer
Put : 1
Intimate consumer
Producer waiting
Consumed : 0
Got : 1
Intimated Producer
Consumed : 1
Put : 2
Intimate consumer
Producer waiting
Got : 2
Intimate Producer
Consumed : 2
Put : 3
Intimate consumer
Producer waiting
Got : 3
Intimate Producer
Consumed : 3
Put : 4
Producer waiting
Got : 4
Consumed : 4
```

Date :

Deadlock

```
Class A {
Synchronized void foo (B b)
String name = Thread.currentThread().getName();
System.out.println(name+" entered A foo");
try {
    Thread.sleep(1000);

catch (Exception e) {
    System.out.println(" A Interrupted");
}
System.out.println(name + " trying to call B last()");
b.last();
}
void last() {
    System.out.println("Inside A.last");
}
}

Class B {
Sychronized void bar (A a) {
String name = Thread.currentThread() getName();
System.out.println(name + " Entered B.bar");
try {
    thread.sleep(1000)
}
catch (Exception e)
{
    System.out.println(" B Interrupted");
}
System.out.println(name + "trying to call A.last()");
a.last();
}
```

Date :

```
void last () {
 System.out.println ("Inside A.last");
  }
 }
Class Deadlock implements Runnable
{
    A a = new A ();
    B b = new B ();
    Deadlock () {
     Thread.currentThread().setName("MainThread");
     Thread t = new Thread(this, "Racing thread");
      t.start ();
      a.foo (b);
      System.out.println ("Back in main Thread")
    }

  public void main () {
      b.bar(a);
      System.out.println ("Back in other thread");
  }
  public static void main (String args []);
  {
    new Deadlock ();
  }
}
```

Output

MainThread extend A.foo
Racing thread extend B.foo
Mainthread trying to call B last ()
Inside A.last
Back in main thread
Receiving thread trying to call A.last ()
Inside A last
Back in other thread.

B.M.S COLLEGE OF ENGINEERING BENGALURU

Autonomous Institute, Affiliated to VTU

LAB REPORT

23CS3PCOOJ

Submitted in partial fulfilment of the requirements for Lab

Bachelor of Engineering

in

Computer Science and Engineering

Submitted by:

NISCHAL KIRAN
1BM22CS182

Department of Computer Science and Engineering, B.M.S

College of Engineering,

Bull Temple Road, Basavanagudi, Bangalore, 560 019 2023-

2024.

INDEX

SL.NO	Title	Date
1	LAB 1	12/12/2023
2	LAB 2	19/12/2023
3	LAB 3	26/12/2023
4	LAB 4	02/01/2024
5	LAB 5	09/01/2024
6	LAB 6	16/01/2024
7	LAB 7	23/01/2024
8	LAB 8	30/01/2024
9	LAB 9	06/02/2024
10	LAB 10	20/02/2024
11	Complete Scanned Observation Book	12/12/202320/02/2024

Lab Programs

1. Develop a Java program that prints all real solutions to the quadratic equation ax 2 +bx+c = 0. Read in a, b,c and use the quadratic formula. If the discriminate b 2 -4ac is negative, display a message stating that there are no real solutions.

```
import        java.util.Scanner;
class quad {
   public static void main(String args[])
      { int a, b, c; double r1, r2, d;
      Scanner s = new Scanner(System.in);
      System.out.println("Nehal A K\n1BM22CS176");
      System.out.println("Enter the coefficients of a,b,c");
      a = s.nextInt(); b = s.nextInt(); c = s.nextInt(); while
      (a == 0) {
         System.out.println("Not  a  quadratic  equation");
         System.out.println("Enter a non zero value for a:");
         a = s.nextInt();
      }
      d = b * b - 4 * a * c;
      if (d == 0) { r1 = (-
      b) / (2 * a);
         System.out.println("Roots are real and equal");
         System.out.println("Root1 = Root2 = " + r1);
      } else if (d > 0) { r1 = ((-b) + (Math.sqrt(d))) /
         (double) (2 * a); r2 = ((-b) - (Math.sqrt(d))) /
         (double) (2 * a); System.out.println("Roots are
         real and distinct");
         System.out.println("Root1 = " + r1 + " Root2 = " + r2);
      } else if (d < 0) {
         System.out.println("Roots are imaginary");
         r1 = (-b) / (2 * a); r2 =
         Math.sqrt(-d) / (2 * a);
         System.out.println("Root1
         = " + r1 + " + i" + r2);
         System.out.println("Root1 = " + r1 + " - i" + r2);
      }

   }
}
```

2. Develop a Java program to create a class Student with members usn, name, an array credits and an array marks. Include methods to accept and display details and a method to calculate SGPA of a student.

```
import java.util.Scanner;
class Subject { int
subjectMarks; int
credits; int grade; }

class Student
  { String
  name;
  String usn; double
  SGPA;
  Subject[] subject;
  Scanner s;



  Student() {
    int i;
    subject = new Subject[9];
      subject[i] = new Subject();
    s = new Scanner(System.in);
  }



  void getStudentDetails()
    { System.out.println("Enter student name");
    name = s.next();
```

```
    System.out.println("Enter student usn"); usn
    = s.next();
  }



  void  getMarks()  {
    int i;
    for (i = 0; i < 9; i++) {
      System.out.println("Enter  marks  for  subject  "  +  (i  +  1));
      subject[i].subjectMarks = s.nextInt();
      System.out.println("Enter  credits  for  subject  "  +  (i  +  1));
      subject[i].credits = s.nextInt();


      if (subject[i].subjectMarks >= 90) {
        subject[i].grade = 10;
      }  else  if  (subject[i].subjectMarks  >=  80  &&
      subject[i].subjectMarks
< 90) { subject[i].grade = 9;

      }  else  if  (subject[i].subjectMarks  >=  70  &&
      subject[i].subjectMarks
< 80) { subject[i].grade = 8;
      }  else  if  (subject[i].subjectMarks  >=  60  &&
      subject[i].subjectMarks
< 70) { subject[i].grade = 7;
      }  else  if  (subject[i].subjectMarks  >=  50  &&
      subject[i].subjectMarks
< 60) { subject[i].grade = 6;
      }  else  if  (subject[i].subjectMarks  >=  40  &&
      subject[i].subjectMarks
< 50) { subject[i].grade = 5;
      } else
        { System.out.println("Failed"
```

```
            ); System.exit(0);
        }
    }
}


void computeSGPA()
    { int totalCredits = 0; int
    creditsGained = 0;
    int i;

    for (i = 0; i < 9; i++) {
        totalCredits += subject[i].credits;
        creditsGained += subject[i].credits * subject[i].grade;
    }

    SGPA = (double) creditsGained / totalCredits;
}


void displayResult()
    {  System.out.println("Name  =  "  +
    name); System.out.println("Usn = " +
    usn); System.out.println("SGPA = " +
    SGPA);
}

public class Main {
```

```
    public static void main(String
      args[]) { Student s1 = new
      Student(); s1.getStudentDetails();
      s1.getMarks();
      s1.computeSGPA();
      s1.displayResult();
    }
  }
```

3. Create a class Book which contains four members: name,author, price, num_pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a toString( ) method that could display the complete details of the book. Develop a Java program to create n book objects. import java.util.Scanner;

```
class Book {
  String name;
  String author;
  int price; int
  numPages;

  Book(String name, String author, int price, int numPages)
    { this.name = name; this.author = author; this.price =
    price; this.numPages = numPages;
  }


  public String toString() {
    String bookDetails = "Book name: " + this.name + "\n" +
        "Author name: " + this.author + "\n" +
        "Price: " + this.price + "\n" +
```

```
            "Number of pages: " + this.numPages + "\n";
    }
}

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

        System.out.println("Enter the number of books: "); int
        n = s.nextInt();



        Book[] books = new Book[n];



        for (int i = 0; i < n; i++) {
            System.out.println("Enter details for Book " + (i + 1) + ":");
            System.out.print("Name: ");
            String name = s.next();
            System.out.print("Author: ");
            String   author   =   s.next();
            System.out.print("Price: ");
            int price = s.nextInt();
            System.out.print("Number of pages: "); int
            numPages = s.nextInt();
            books[i] = new Book(name, author, price, numPages);
        }
        System.out.println("\nDetails of the books:"); for
        (int i = 0; i < n; i++) {
```

```
            System.out.println("Book " + (i + 1) + ":\n" + books[i].toString());
        }
    }
}
```

4. Develop a Java 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.

```
import java.util.*;

abstract  class  AbsArea  {
   int a, b;

   AbsArea(int x)
      { a = x;
   }

   AbsArea(int x, int y) {
      a = x;
      b  =  y;
   }

   abstract void area();
} class rec extends

AbsArea { rec(int a, int b)

{

      super(a,   b);
   }

   void area() {
      System.out.println("The area of the rectangle is: " + a * b); }
}
```

```
class tri extends AbsArea {
   tri(int a, int b) {
      super(a, b);
   }

   void area() {
      System.out.println("The area of the triangle is: " + (a * b) / 2);
   }
}

class cir extends AbsArea {
   cir(int a) {
      super(a);
   }

   void area() {
      System.out.println("The area of the circle is: " + 3.14 * a * a);
   }
}

class Main {
   public static void main(String args[])
{ System.out.println("This is done by Nehal
AK\n1BM22CS176");
      int x, y;
      Scanner n = new Scanner(System.in);

      // Input for rectangle dimensions (x and y) with validation
      x = n.nextInt(); y
      = n.nextInt(); if
      (x < 0 || y < 0) {
         System.out.println("Invalid input for rectangle. Please enter
positive values.");
         // You might want to handle this situation differently, such as asking
the user to enter values again.
         System.exit(1); // Exiting with status code 1 (indicating abnormal
exit)
      }

      AbsArea r = new rec(x, y);
      r.area();
```

```
        // Input for triangle dimensions (x and y) with validation
        System.out.println("Give input for triangle");
        x = n.nextInt(); y = n.nextInt(); if (x < 0 || y <
        0) {
            System.out.println("Invalid input for triangle. Please enter positive
    values.");
            System.exit(1);
        }

        AbsArea t = new tri(x, y); t.area();

        // Input for circle radius (x) with validation
        System.out.println("Give input for circle");
        x = n.nextInt(); if (x < 0) {
            System.out.println("Invalid input for circle. Please enter a positive
    value.");
            System.exit(1);
        }

        AbsArea c = new cir(x);

        c.area();
    }
}
```

5.Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed.

```
import      java.util.Scanner;

class Account {

   String CustomerName; double

   AccNo, Balance;
```

```
    Account(String CustomerName, double AccNo, double Balance)
        { this.CustomerName = CustomerName; this.AccNo = AccNo;
        this.Balance = Balance;
    }

    public void Deposit(double Amount) {
        Balance += Amount;
        System.out.println("DEPOSIT                    SUCCESSFUL");
        DisplayBalance();
    }
    void DisplayBalance() {
        System.out.println("BALANCE:" + Balance);
    }
}

class CurrentAccount extends Account {
    double MinBalance = 500.0; double
    Charges = 10.0;

    CurrentAccount(String CustomerName, double AccNo, double Balance) {
        super(CustomerName, AccNo, Balance);
    }

    void Withdraw(double Amount) {
        if (Balance >= Amount) {
            Balance -= Amount;
            System.out.println(Amount + " withdrawn successfully");
            DisplayBalance();
        } else {
```

```
        System.out.println("insufficient Balance");
    }
  }

  void   UpdateBalance()   {   if
    (Balance <= MinBalance) {
        Balance -= Charges;
        System.out.println("service   charge   applied   for   maintaining   low
balance");
        DisplayBalance();
    }
  }
}

class SavingsAccount extends Account {
  SavingsAccount(String CustomerName, double AccNo, double Balance) {
    super(CustomerName, AccNo, Balance);
  }

  double interest = 0.05;

  void UpdateBalance() {
    Balance = Balance + (interest * Balance);
    DisplayBalance();
  }

  void Withdraw(double Amount) {
    if (Balance >= Amount)
        { Balance -= Amount;
        DisplayBalance();
```

```
        } else {
            System.out.println("insufficient Balance");
        }
    }
}

class Bank {
    public static void main(String args[])
        {  String  CustomerName;  double
        AccNo, Balance; double amt, amt1;
        Scanner in = new Scanner(System.in);

        System.out.println("enter name:");
        CustomerName = in.next();

        System.out.println("enter AccNo:");
        AccNo = in.nextDouble();
        System.out.println("enter Balance:");
        Balance = in.nextDouble();
        System.out.println("enter amount to deposit"); amt =
        in.nextDouble();
        System.out.println("enter amount to withdraw"); amt1
        = in.nextDouble();

        CurrentAccount c = new CurrentAccount(CustomerName, AccNo,
Balance);
        c.Deposit(amt);
        c.Withdraw(amt1);
        c.UpdateBalance();
```

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


        SavingsAccount s = new SavingsAccount(CustomerName, AccNo,
Balance);
        s.Deposit(amt);
        s.Withdraw(amt1);
        s.UpdateBalance();
    }
}
```

6.Create a package CIE which has two classes- Student and Internals. The class Student has members like usn, name, sem. The class Internals derived from Student has an array that stores the internal marks scored in five courses of the current semester of the student. Create another package SEE which has the class External which is a derived class of Student. This class has an array that stores the SEE marks scored in five courses of the current semester of the student. Import the two packages in a file that declares the final marks of n students in all five courses.

```
    package CIE;
    public class
    Student
    { public String name;
        public String usn;
        public int sem;

    public Student(String name,String usn,int sem)
      { this.name=name;
        this.usn=usn;
        this.sem=sem;
      }
    }
  package CIE;


  public class Internals extends CIE.Student
  {
```

```
        public int [] InternalMarks;

        public Internals(String name , String usn , int sem , int []InternalMarks)
    { super(name , usn , sem);
        this.InternalMarks=InternalMarks;
    }  }  package  SEE;
    import
    CIE.Student;

    public class Externals extends Student
      {
          public int [] SeeMarks;

          public Externals(String name , String usn , int sem , int []SeeMarks)
           { super(name , usn , sem);
               this.SeeMarks=SeeMarks;
           }
      }
import        CIE.Student;
import       CIE.Internals;
import      SEE.Externals;
import java.util.Scanner;


public class FinalMarks
{ public static void main(String [] args)
      {
          Scanner s1=new Scanner(System.in);

         System.out.println("Enter  the  number  of  Students");  int
         n=s1.nextInt();
       String []names=new String[n];
         String []usn=new String[n]; int []sem
       = new int[n]; int [][] InternalMarks =
       new int[n][5];
      int [][] SeeMarks = new int[n][5];

for(int i=0 ; i<n; i++)
{
      System.out.println("Enter details for Student" + (i+1) + ":");
      System.out.println("Name:");
      names[i]=s1.next();
      System.out.println("USN:"); usn[i]=s1.next();
      System.out.println("SEM:"); sem[i]=s1.nextInt();

      System.out.println("Enter Internal marks for 5 courses:"); for(int j=0;
j<5; j++)
{
```

```
        System.out.println("Course"+(j+1)+":");
        InternalMarks[i][j]=s1.nextInt();
}
     System.out.println("Enter Internal marks for 5 courses:"); for(int j=0;
j<5; j++)
{
    System.out.println("Course"+(j+1)+":");
    SeeMarks[i][j]=s1.nextInt();
}
}

int [][]FinalMarks =  new int[n][5];
for(int i=0 ; i<n ; i++)
{
          Internals I1 = new Internals(names[i] , usn[i] , sem[i] , InternalMarks[i]); Externals
          E1 = new Externals(names[i] , usn[i] , sem[i] , SeeMarks[i]);


for(int j=0; j<5 ;j++)
{
          FinalMarks[i][j] = I1.InternalMarks[i] + E1.SeeMarks[j];
}
          System.out.println("Finals Marks for " + n+ "Students in 5 courses:"); for(i=0 ;i<n
;i++)
{
          System.out.println(names[i] +":");

for(int j=0; j<5;j++)
{
          System.out.println(FinalMarks[i][j] + ":");
}
         System.out.println();
        } s1.close();
} }
}
```

7. Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called “Father” and derived class called “Son” which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge( ) when the input age<0. In Son class, implement a constructor that cases both father and son’s age and throws an exception if son’s age is >=father’s age.

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

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

}

class InputScanner {
  static Scanner sc = new Scanner(System.in);
}

class Father extends InputScanner {
  int fatherAge;

  Father() throws WrongAge
    { System.out.println("Enter Father's
    age"); fatherAge = sc.nextInt(); if
    (fatherAge < 0) {
      throw new WrongAge("Age cannot be negative");

    }

  }

  void display() {
    System.out.println("Father's age is " + fatherAge);

} }

class Son extends Father {
  int sonAge;

  Son() throws WrongAge
    { System.out.println("Enter son's
    age"); sonAge = sc.nextInt(); if
    (sonAge > fatherAge) {
        throw new WrongAge("Son's age cannot be greater than father's age");
    } else if (sonAge < 0) { throw new WrongAge("Age
    cannot be negative"); }

  }

  void display() {
    System.out.println("Son's age is " + sonAge);
```

```
    }

}

public class ExceptionHandling {
    public static void Main(String args[]) {
        try{
            Son son= new Son();
            son.display();
        }
        catch(WrongAge e){
            System.out.println("Exception "+ e.getMessage());
        }

    }
    }
```