Date 12/12/23
Page

1) Quadratic Equation

#Program.

```
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: ");
            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);
        }
```

classmate
Date
Page
12/12/23

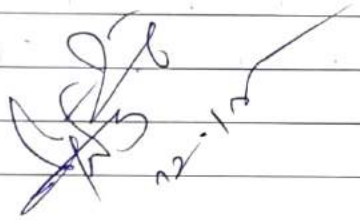
```
else if (d>0)
{
    r1 = ((-b) + (Math.sqrt(d)))/(double)(2*a);
    r2 = ((-b) - (Math.sqrt(d)))/(double)(2*a);
    System.out.println("Roots are real and distinct");
    System.out.println("Root1 =" + r1, "Root2 =" + r2);
}
else if (d<0)
{
    System.out.println("Roots are imaginary");
    r1 = (-b)/(2*a);
    r2 = Math.sqrt(-d)/(2*a);
    System.out.println("Root 1 =" + r1 + "+i" + r2);
    System.out.println("Root 2 =" + r1 + "+i" + r2);
}
}

class Quadratic Main
{
    public static void main(String args[])
    {
        Quadratic q = new Quadratic();
        q.getd();
        q.compute();
    }
}
```

OUTPUT:-

Enter Coefficients of a, b, c
4, 5, 6
Roots are imaginary
Root 1 = 0.0+i 0.5826872164704049
Root 2 = 0.0-i 0.5826872164704049

Enter the coefficients of a, b, c.
1 -2 1
Roots are real and equal
Root 1 = Root 2 = 1.0

Enter the coefficients of a, b, c.
1 -3 2
Roots are real and distinct
Root 1 = 2.0 Root 2 = 1.0



classmate
Date 19/12/23
Page

LAB-2 program

```
import java.util.Scanner;

class Student {
    private String USN;
    private String name;
    private int[] credits;
    private int[] marks;

    public Student (String USN, String name, int[]
                    credits, int[] marks) {
        this.USN = USN;
        this.name = name;
        this.credits = credits;
        this.marks = marks;
    }
    public void accept details () {
        Scanner sc = new Scanner (System.in);
        System.out.println ("Enter USN:");
        USN = sc.nextLine ();
        System.out.println ("Enter Name : ");
        name = sc.nextLine ();
        System.out.println ("Enter the number
            of subjects : ");
        int numSubjects = sc.nextInt ();
        credits = new int [numSubjects];
        marks = new int [numSubjects];
        for (int i=0; i<numSubjects; i++) {
            System.out.println ("Enter the
            credits for subjects "+(i+1)+" : ");
            credits [i] = sc.nextInt ();
            System.out.println ("Enter the mark
            for subject "+(i+1)+" : ");
            marks [i] = sc.nextInt ();
```


3

```
public void displayDetails(){
    System.out.println("USN: "+USN);
    System.out.println("Name: "+name);
    for(int i=0; i<credits.length; i++){
        System.out.println("Subject"+(i+1)
        +": Credits: "+credits[i]+" Mark: "+
        marks[i]);
    }
}

public double calculateSGPA(){
    double totalCreditPoints=0;
    double totalCredits=0;
    for(int i=0; i<credits.length; i++){
        totalCreditPoints+=credits[i]*getGrade
        GradePoints(marks[i]);
        totalCredits+=credits[i];
    }
    return totalCreditPoints/totalCredits;
}
private double getGradePoints(int marks){
    if(marks>=90){
        return 10.0;
    }else if(marks>=80){
        return 9.0;
    }else if(marks>=70){
        return 8.0;
    }else if(marks>=60){
        return 7.0
    }else if(marks>=50){
        return 6.0
    }else{
        return 0.0;
}
}
```

```
public static void main (String[] args){
    Student = new Student ("4B
    student.acceptdetails();
    student.displaydetails();
    System.out.println("SGPA: "+ student.
    calculatesgpa());
    }
}
```

OUTPUT:

Enter USN:
1BM22CS229
Enter name:
Rakhal.
Enter number of Subjects
3
Enter credit for Subjects 1
3
Enter mark for subjects 1
90
Enter credit for Subject 2
4
Enter marks for Subject 2
90
Enter credit for Subject 3
4
Enter marks for subject 3
85

SGPA: 9.6363

LAB - 3 PROGRAMS

8) Create a class Book which contains its members: name, author, price, num_pages. Include a constructor to set the values of the members. Include methods to set and

```
import java.util.Scanner;
class books {
        String author, book;
        int price, numpages;

        books (String book, String author,
        price, int numpages)
        {
            this.book = book;
            this.author = author;
            this.price = price;
            this.numpages = numpages
```

```
public String toString ():
{
    return "Book Name: "+this.name+"\n Author:
    author + "\n price: "+this.price"\n pa
    ber"+this.numpages.;
}
```

```
Class lib {
        String author, book;
        int price, numpages;
        int num_books;
        Scanner input = new Scanner (System.
        num_books = input.nextInt ()
        books b[] = new books [num_book
```

```
for (int i = 0; i<num_book; i++){
        input.nextline();
        System.out.println("Enter bookname:");
        book = input.nextline();
        System.out.println("Enter author
        name");
        author = input.nextline();
        System.out.println("Enter price");
        price = input.nextInt();
        System.out.println("Enter p-num");
        numpages = input.nextInt();
        b[i] = new books(book, author, price,
        numpages);
    }
    for(i=0; i<numpages; i++){
        String a = b[i].toString();
        System.out.println(a);
    }
}
```

Output :-

```
Book 1
Enter book name : Harry Potter
Enter author : JK Rowling
enter price : 799
enter pgnum : 885

Book 1 details
Book Name : Harry Potter.
Author : JK Rowling.
Price: 799
Pages num : 889
```

Lab 4 program:

5)

Input:-

```
import java.util.Scanner;

class inputScanner {
    void rec(rectangle ab){
        Scanner input
```

```
abstract class shape extends inputScanner{
    int a, b;
    abstract void printArea();
}
```

```
class rectangle extends shape{
    rectangle(){
        rec(this);
    }
    void printArea(){
        System.out.println("Area of
        (double)(a*b));
    }
```

```
class Triangle extends Shape {
    public Triangle(int base, int height
    triangle (tri(this);}
    void printArea () {
            System.out.println("Area of
        triangle = "+(0.5*a*b);
    }
}

class circle extends shape {

    triangle circle (){cir(this);}
    void printArea (){
            System.out.println("Area of
        the circle = "+(3.14*a*a));
        }
}

class calc {
    public static void main(String[] args){
    rectangle r = new rectangle ();
    triangle t = new triangle ();
    circle c = new circle ();

    r.printArea ();
    t.printArea ();
    c.printArea ();
    }
}
```

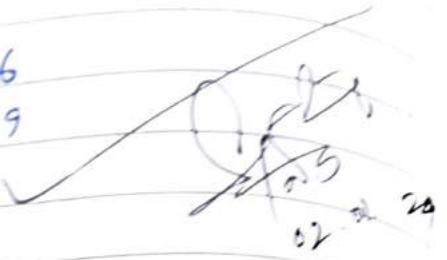
OUTPUT:

Enter the dimension of the rectangle (Length & Breadth):

4 5

Enter the dimension of triangle (ba...
5 9
Enter the dimension of circle (radius)
6

Area of Rectangle: 20
Area of triangle: 36
area of circle = 113.0399



Lab 5

```
import java.util.Scanner

class Account {
    String name;
    String Accno;
    boolean current;
    double balance = 0;
    int main balance = 0;
    Scanner sc = new Scanner (System.in);
    Account () {
        if (this.getclass() = current Acc class
            current = true;
    }
    else {
        current = false;
    }

    name = sc.next();
    System.out.println("Enter account no.:");
    acc_no = sc.nextInt();
}
void deposit () {
    System.out.println ("Enter deposit amount
    );
    balance += sc.nextDouble();

void withdrawal () {
    System.out.println ("Enter deposit amount);
    if (amount > this.balance) {
        System.out.println("not enough
    balance");
    }
```

```
        else {
            this.balance -= amount;
        }

    void showBalance() {
        System.out.println("Balance = " + balance);
    }

void CurrentAcc extends Account {
    void cheque() {
        System.out.println("Enter cheque amount: ");
        double cheque = sc.nextDouble();
        withdraw(cheque);
        System.out.println("Cheque cleared");
    }
}
class SavingsAcc extends Acc {

    void compound(int p, int t) {
        balance = balance * (Math.pow((1 + (double)(r/100)), t));
        System.out.println("Balance after given
        date and time = " + balance);
    }
}
class Bank {
    public static void main(String[] args) {
        SavingsAcc john = new SavingsAcc();
        CurrentAcc Smith = new CurrentAcc();
        Account ref = null;
        System.out.println("1. ... men ...");
        System.out.println("1. Deposit 2. Withdraw
        3. Compound interest");
```

```
\n4 Display Account details
\n5 create cheque \n6 Exit \n choice");

choice = sc.nextInt();
System.out.println("Enter account no:");
acc = sc.nextInt();
if(acc == 1){
    ref = joun;
}
else{
    ref = smith;
}
while(choice != 6){
    ref.deposit();
}
elseif(choice == 2){
    ref.withdrawal();
}
elseif(choice == 3){
    if(acc == 1){
        john.compound(1,5);
    else if(choice == 4)
        ref.showbalance();
}

System.out.println("Enter acc no.");
acc = sc.nextInt();
choice = sc.nextInt
```

OUTPUT:-

Enter account name : John
Enter account number : 1

..... MENU .....

1) Deposit
2) withdraw
3) compute interest for savings A.c
4) Display account details
5) create cheque
6) Exit
Choice : 1.
Enter account no. 1
1
Enter deposit amount : 100
Enter account no. : 2
..... Menu .....

1) Deposit
2) withdraw
3) compute interest for savings acc.
4) display account details
5) create cheque
6) Exit
choice 1
enter account no : 1
Enter deposit amount : 100
Enter acc. no. : 2
Enter choice : 6

09.01.24

Lab-6

program 1 output :-

OUTPUT:-

Empty string :
String from char Array: hello
Substring from char array : orl
copied string copy me !!

program 2 output.

OUTPUT:-

String length is : 12
String literal - java is fun.
Concatenated string HelloWorld.

program 3 output

Demo object with heartbeat 72 beats per minute.

program 4 output:

Extracted string : BMSCE

program 5 output :-

output using getBytes
Bytes : 72 101 108 108 111 44 32 87 111
114 108 100 33

Reconstructed string : Hello world!

outputs for toCharArray():

characters : programming
Reconstructed String : Programming.

(6) program output for 6th question is:-

Bmsce equals Bmsce → true
Bmsce equals College → false
Bemsce equals Bmsce → false
Bmsce equalsIgnoreCase BMSCE → true.

(7) program output for 7th Question is

Substring is matched.

(8) program output for 8th Question is

Starts with Hello : true
Starts with java : false

(9) program output for 9th Question is

Ends with world! : true
Ends with Java : false.

(10) program output for 10th Question is

using equals(): true.
using == for different objects: false
using != for same objects : true

(11) program output for 11th program.

Sorted words :

apple
ball
cat
dog
ent
free
gun
hen
ice
jug
kite
lift
man
net
orange
parrot
queen
ring
star
tree
umbrella
van
watch
xmas
yatch
zee.

12) Output for 12th Program

Sorted numbers

1
2
3
4
5
6
7
8
9
10.

13) output for 13th program

Modified string : This is a test. This is too

14) output for 14th program

Concatenated string : helloworld.

15) output :

modified string : Welcome to Bmsce Commge of Engine
-ng

16) output :

Trimmed string : Hello friends.

18) Output:

After setLength (5): Hello
charAt (2): l
After setCharAt (2, 'x'): Hexlo
getChars (0, 5, targetArray, 0): Hexlo
After append: Hexlo Appended
After insert (7, 'Inserted'): Hexlo
Inserted Appended!
After reverse: !dednappA detrepsnI
olxeH
After delete (3, 9): !drtednappA detrepsnI
olxeH
After deleteCharAt (0): drtednappA detrepsnI
olxeH
After replace (0, 5, 'New'):
New drtednappA detrepsnI olxeH
substring (3, 8) dtedn.

19) Output:-

Eagle Behavior
Eagle soars the sky
Eagle screeches loudly

Hawk behaviors.
Hawk glides gracefully in the air
Hawk emits a high pitched scream.

20 Output:
circle - Area: 78.53981633974483,
perimeter: - 31.415926535897933
Triangle - Area - 6.0; perimeter: 12.0.

LAB-6

8) Create a package CIE which has 2 classes Students and Internals. The class Students has member like USN, name, sem. The class Internals derived from Students has an array that stores the internal marks scored in 5 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 5 courses of the current semester of the student. Import the two packages in a file that declares the final marks of n students of all 5 courses.

Input:

in file student.java

```
package CIE;
import java.util.Scanner;
public class Student{
    protected String usn = new String();
    protected String name = new String();
    protected int sem;

    public void inputStudentDetails(){
        Scanner scanner = new Scanner(System.in);

        System.out.println("Enter USN: ");
        usn = scanner.nextLine();
        System.out.println("Enter name");
        name = scanner.nextLine();
        System.out.println("Enter semester");
        sem = scanner.nextInt();
}
```

```
public void displayStudentDetails(){
        System.out.println("USN: "+usn);
        System.out.println("Name: "+name);
        System.out.println("Semester: "+sem);
}
}
```

Internals file.java.

```
package CIE;

import java.util.Scanner;

public class Internals extends Student{
        protected int marks[] = new int[5];

        public void inputCIEMarks(){
                Scanner scanner = new Scanner
                (System.in);

                System.out.println("Enter CIE marks for "+
                name + ":");
                for(int i=0; i<5; i++){
                  System.out.println("Enter marks for
                  Subject "+(i+1)+":");
                  marks[i] = scanner.nextInt();
                }
        }
}
```

External.java final.

P.T.O.

```
package SEE;

import CIE.Internals;
import java.util.Scanner.

public class Externals extends Internals{
        protected int marks[];
        protected int finalmarks[];

        public Externals(){
            marks = new int[5];
            finalMarks = new int[5];
        }

        public void inputSEEmarks(){
            Scanner scanner = new Scanner(System.
                                          in);

            System.out.println("Enter SEEmas
            -ks for "+name+":");
            for(int i=0; i<5; i++){
                System.out.println("Enter
            marks for subject "+(i+1)+":");
            marks[i] = scanner.nextInt();
        }
    }
}

public void calculateFinalMarks(){
    for(int i=0; i<5; i++){
        finalMarks[i] = marks[i]/2 +super.
        marks[i];
    }
}
```

```
public void displayFinalMarks(){
    displayAllStudentDetails();
    System.out.println("Final marks");
    for(int i=0; i<5; i++){
        System.out.println("Subject "+(i+1)+":"+
        finalMarks[i]);
    }
}
```

Main.java

```
import SEE.Externals;

public class Main{
    public static void main(String args[]){
        int numOfStudents=2;
        Externals finalMarks[] = new Externals
        [noOfStudents];

        for(int i=0; i<numOfStudents; i++){
            finalmarks[i] = new Externals();
            finalMarks[i].inputStudentdetails()

        for(int i=0; i<numOfStudents; i++){
            finalMarks[i] = new Externals();
            finalmarks[i].input.Student
            Details();

            System.out.println("Enter CIE marks
            for "+finalmarks[i].name);

            finalMarks[i].inputCIEMarks();

            System.out.println("Enter the SEE
            Marks"); }
```

finalMarks[i].inputSEEmarks();
}

System.out.println("Displaying data:\n");

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

finalMarks[i].calculateFinalMarks();

finalMarks[i].displayFinalMarks();
}
}
}

OUTPUT:-

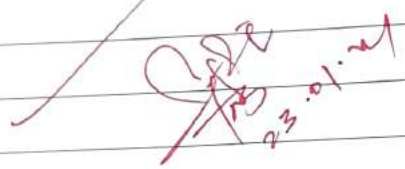
Enter the no. of students.

1
Enter the USN
1BM22CS229

Enter the name.
S Rathod
Enter the Semester
3
Enter the marks for subject 1:
50
Enter the marks for subject 2:
50
Enter the marks for Subject 3:
50
Enter the marks for Subject 4:
50

Enter the marks for Subject 5:
50
Enter SEE marks
Subject 1 marks: 50
Subject 2 marks: 50
Subject 3 marks: 50
" 4 " : 50
" 5 " : 50

Displaying data.
Name: S Rakshak
Subject 1: 75
" 2: 75
" 3: 75
" 4: 75
" 5: 75



LAB-7

classmate
Date 30/1/2024
Page

8) write a program that demostrates 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 age and throws an exception if son's age >= father's age.

```
import java.util.Scanner;

class WrongAge extends Exception {

    public WrongAge() {
        super("Age Error");
    }

    public WrongAge(String message) {
        super(message);
    }
}

class InputScanner {
    protected Scanner scanner;

    public InputScanner() {
        this.scanner = new Scanner(System.in);
    }
    public int getIntInput() {
        return scanner.nextInt();
    }
}
```

```
class Father extend Input Scanner {

    protected int father age

    public Father () throws WrongAge {
        System out . println ("Enter father's
    age ");
    fAther Age = get Int Input ();

    if (father Age <= 0) {
        throw new wrongAge (." Age
    cannot be negative");
    }
}

public void display () {
    System.out.println ("Fathers age : " + fatherAge
    ;
    }
}

class Son extend father {

    private int Son Age;

    public Son () throws Wrong Age {
        super();
        System.out.println ("Enter son's age: ");
        son Age = get Int Input ();

        if (sonAge >= father Age) {
            throw new Wrong Age ("Son's age
cannot be greater than fathers age ");
} else if (son Age <= 0) {
```

```
        throw new wrongAge ("Son's age cannot be
greater than father's age");
        else if (sonAge<0)
            throw new wrongAge ("Age cannot be given
negative");
        }
}

public void display () {
        super.display ();
        System.out.println ("Son's age: " + sonAge);
    }
}

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

O.P:-

Enter father's age :- 1

Error : Age cannot be negative

Ender fathers age: 45

Enther sons age : 0

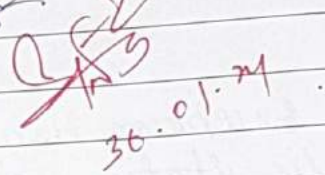
Error: Age cannot be given negative

Enter fathers age 45
Enter sons age 55

Error son's Age cannot be greater than father age.

Enter fathers age : 45
Enter son's age : 25

fathersage : 45
son's age : 25



30.01.24

Week 8

8) write a program which creates 2 threads, one thread displaying "BMS College of Engineering" once every 10 seconds and another displaying "CSE" once every 2 seconds.

I/P:

```
class DisplayThread extends Thread {
    private String message;
    private int delay;
    private int repetations;

public DisplayThread (String message, int delay,
int repetations) {
    this.message = message;
    this.delay = delay;
    this.repetitions = repititions;
}
public void run() {
    for (int j=0; i< repitition; i++) {
        System.out.println(message);
        try {
            Thread.sleep(delay * 1000);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
}
}

public class ThreadExample {
public static void main (String args [ ]) {
for int (i=0; repitition < 5; repitition++) {
    DisplayThread bms Thread = new Display
```

```
Thread C" BMS college of Engineering, 10);

bms Thread.start();

try {
    bms Thread.join();
} catch (Interrupted Exception e) {
        e.printStackTrace();
}
Display Thread csc Thread = new Display
-Thread("CSE", 2, 5);

cse Thread.start();

try {
    cseThread.join().
} catch (Interrupted Exception e){
        e.printStackTrace();
    }
}
}
```

OUTPUT:

BMS college of Engineering
CSE
CSE
CSE
CSE
CSE

write one more

06.02.24

classmate
Date
Page

Inter process communication

Input / code:

```
class S {
int n;
boolean valueset = false;
synchronized int get() {
while (! value set )
try {
System. out . println ("\n Consumer waiting\n")
wait();
} catch (Interrupted Exception caught ");
}
System.out.println ("Got: "+n);
synchronized
    valueset = false ;
    System.out.println ("\n Intimate producer");
    notify ();
    return n;
}

Synchronized void put ( int n){
while (valueset )
try {
System out println ("\n producer waiting \n");
wait ();
} catch (Interrupted Exception e) {
System.out . println ("Interrupted Exception
caught ");
}
this. n = n;
valueset = true;
System.out.println ("\n Intimate Consumer\n");
```

```
noteify();
}
}

class producer implements Runnable {

Q q;

producer (Q q) {
this.q = q;
new Thread (this, "producer"). start ();
}
public 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:" + q);
i++;
}
}
}
class PCFixed {
    public static void main (String args[])
    Q q = new Q();
    new producer (q);
    new consumer (q);
```

System.out.println("press Control-C to stop");
}
}

OUTPUT

Put : 1
Got : 1
put : 2
Got : 2
Put : 3
Got : 3
Put : 4
Got : 4
put 5
Got 5

DEBUG DEADLOCK

Input

→

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

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

OUTPUT :-

```
Main Thread entered A.foo
Racing Thread entered B.bar
Main Thread trying to call B.last()
Inside A.last
Back in main thread
Racing Thread trying to call A.last()
Inside A.last
Back in other thread.
```

Week 9

9) Write a program that creates a user interface to perform integer divisions. The user enters 2 numbers. The div. of no. 1 & no. 2 is displayed in the Result field. when div. button is clicked. If the entered no. is not an int then throw number format Exception. If num 2 is 0. the program would throw an ArithmeticException.

code / Input:

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

class SwingDemo {
    SwingDemo(){
    // create jframe container
    JFrame jfrm = new JFrame("Divider App");
    jfrm.setSize(275, 150);
    jfrm.setLayout(new FlowLayout());
    jfrm.setDefaultCloseOperation(JFrame.EXIT_
    _CLOSE);

    // text label
    JLabel jlab = new JLabel("Enter the divider
    and dividend:");

    JTextField ajtf = new JTextField(8);
    JTextField bjtf = new JTextField(8);

    JButton button = new JButton("Calcula
    te");
```

P.T.O.

// labels.

```
JLabel err = new JLabel();
JLabel alab = new JLabel();
JLabel blab = new JLabel();
JLabel anslab = new JLabel();

// add in order.

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 l = new ActionListener(){
    public void actionPerformed(ActionEvent ev
    {
      System.out.println("Action event from a
      text field");
      }
  };
  ajtf.addActionListener(l);
  bjtf.addActionListener(l);

  button.addActionListener(new ActionListener
{
    public void actionPerformed(ActionEvent ev
    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 Integers!");
    }
    catch (ArithmaticException e){
        alab.setText(" ");
        blab.setText(" ");
        anslab.setText(" ");
        err.setText("B should be NON zero!");
    }
  }
});

  //display frame.
  jfrm.setVisible(true);
  }

public static void main(String args[]){

    SwingUtilities.invokeLater(new Runnable(){
        public void run(){
            new SwingDemo();
        }
    });
}
}
```

OUTPUT:-

Enter the divider & dividend
4 4
calculate A=4 B=4 Ans=1

Enter only Integers!
Enter the divider & divident
int 4
calculate

B should be NON ZERO!
Enter the divider and divident:
3 0
Calculate

Report on all the functions used.

1) JFrame: 'JFrame' is the main window containes for the GUI. It is responsible for the overall structure of the application window, including its title size, layout manager, and default close operation.
2) JLabel: 'JLabel' is used to display text on the GUI. In this program, labels care used to prompt the user to enter divider and dividend, and to display additional information or error messages.
3) Set size :- In java GUI programming, the set size function is used to specify the dimensions of a graphical content such as a window, panel or button. It is typically used to define the width and height of the component for eg: set size (275, 150) set width to 275 pixels and height to 150 pixels.

→ set layout() meathod. Common layouts include Border layout, Flow Layout, GridLayout etc. Example: setLayout(new BorderLayout()); assigns BorderLayout to the container, arranging components accordingly.

→ Set default close operation(): In Java GUI programming to set the default close operation for a JFrame, you can use the 'setDefaultCloseOperation()' meathod. In the above example '(JFrame.EXIT_ON_CLOSE)' sets the default close operation to exit the application when the JFrame is closed. There are a few other options available based on the desired behaviour.

→ To create a 'JTextField': it creates a text field that can hold up to 20 characters wide. You can adjust the width according to your requirements. Then 'add(textField)' adds the text field to the JFrame.

→ Afterwards add frame a JFrame is created the a jTextfild is created. add frame function allows you to add frames to youre main JFrame

→ Action Listener: In Java GUI programming, an action listener is an interface used to handle events triggered by user actions, such as clicking a button or selecting an item from a dropdown menu. When an event occurs, the corresponding ActionListener's actionPerformed is invoked this allows you to define what action should be taken in response to the users interaction

→ setText: In Java 'setText()' is a meathod used to set the text content of a text based component, such as JLabel, 'JButton', 'JTextField' 'JTextArea'.

Lab 1

```java
import java.util.Scanner;

class QuadraticEquation {
   public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        double a, b, c;
        double D;
        double r1, r2;

        System.out.print("Enter a, b, & c: ");
        a = s.nextDouble();
        b = s.nextDouble();
        c = s.nextDouble();

        D = b * b - 4 * a * c;

        if (D >= 0) {
            r1 = (-b + Math.sqrt(D)) / (2 * a);
            r2 = (-b - Math.sqrt(D)) / (2 * a);
            System.out.println("r1 = " + r1);
            System.out.println("r2 = " + r2);
        } else {
            r1 = -b / (2 * a);
            r2 = Math.sqrt(-D) / (2 * a);
            System.out.println("r1 = " + r1 + "+" + r2 + "i");
            System.out.println("r2 = " + r1 + "-" + r2 + "i");
        }
   }
}
```

Lab 2

```
import java.util.Scanner;

class Subject {
   String subjectMarks;
   int credits;
   int grade;
}

class Student {
   Scanner s = new Scanner(System.in);
   Subject[] subject = new Subject[8];
    String name, usn;
   double sgpa;

   Student() {
       for (int i = 0; i < 8; i++) {
           subject[i] = new Subject();
       }
   }

   void getStudentDetail() {
       System.out.print("Enter student name: ");
       name = s.nextLine();
       System.out.print("Enter student usn: ");
       usn = s.nextLine();
   }

   void getMarks() {
       for (int i = 0; i < 8; i++) {
           System.out.print("Subject" + (i + 1));
           System.out.print("Enter marks: ");
           subject[i].marks = s.nextInt();
```

```
            System.out.print("Enter credits: ");
            subject[i].credit = s.nextInt();

            int t = subject[i].subject/10;
            if(t < 4)
              t = 0;
            t = t + 1;
            if(t > 10)
              t = 10;
            subject[i].credits = t;
        }
    }

    void computeSGPA() {
        int tg = 0;
        double tc = 0;
        for (int i = 0; i < 8; i++) {
            tc += subject[i].credits;
            tg += subjects[i].grade * subjects[i].credit;
        }
        sgpa = tg / (tc * 1.0);
        System.out.println("Your SGPA is: " + sgpa);
    }
}

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

Lab 3

```
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 v = "";
       v += "Book name: " + this.name + "\n" +
         "Author name: " + this.author + "\n" +
         "Price: " + this.price + "\n" +
         "Number of pages: " + this.numPages;
       return v;
   }
}

public class Main {
   public static void main(String[] args){
       int n;
       Book[] b;
       String name, author;
       int price, numPages;
```

```
        Scanner s = new Scanner(System.in);
        System.out.print("Enter the number of books: ");
        n = s.nextInt();
        b = new Book[n];

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

        System.out.println("\nDisplaying book details: ");
        for (int i = 0; i < n; i++) {
            System.out.println(b[i].toString());
        }
    }
}
```

Lab 4

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

class InputScanner {
 Scanner s;
  InputScanner() {
   s = new Scanner(System.in);
 }
  public int takeInput(String m) {
   System.out.println(m);
   return s.nextInt();
 }
}

abstract class Shape {
   int dim1, dim2;

   Shape(int dim1, int dim2) {
       this.dim1 = dim1;
       this.dim2 = dim2;
   }

   public abstract void printArea();
}

class Rectangle extends Shape {
   Rectangle(int length, int breadth) {
       super(length, breadth);
   }

   public void printArea() {
```

```
        System.out.println("Area of rectangle = " + (1.0 * dim1 *
dim2));
    }
}

class Triangle extends Shape {
    Triangle(int base, int height) {
        super(base, height);
    }

    public void printArea() {
        System.out.println("Area of triangle = " + (0.5 * dim1 *
dim2));
    }
}

class Circle extends Shape {
    Circle(int radius) {
        super(radius, radius);
    }

    public void printArea() {
        System.out.println("Area of circle = " + (3.1415 * dim1 *
dim2));
    }
}

public class Main {
    public static void main(String[] args) {
        InputScanner ic = new InputScanner();
        Shape shape1, shape2, shape3;
        int d1, d2;

        System.out.println("Rectangle");
        d1 = ic.takeInput("Enter length:");
```

```
        d2 = ic.takeInput("Enter breadth:");
        shape1 = new Rectangle(d1, d2);
        shape1.printArea();

        System.out.println("Triangle");
        d1 = ic.takeInput("Enter base:");
        d2 = ic.takeInput("Enter height:");
        shape2 = new Triangle(d1, d2);
        shape2.printArea();

        System.out.println("Circle");
        d1 = ic.takeInput("Enter radius:");
        shape3 = new Circle(d1, d1);
        shape3.printArea();
    }
}
```

Lab 5

```
import java.util.Scanner;

abstract class Account {
   String customerName;
   long accountNumber;
   String accountType;
   double balance;
   boolean updated = false;

   public Account(String customerName, long accountNumber, String
accountType, double balance) {
       this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.accountType = accountType;
        this.balance = balance;
   }

   public void deposit(double amount) {
        updated = false;
        balance += amount;
        System.out.println("Deposit of ₹ " + amount + " successful.");
   }

   public void withdraw(double amount) {
         updated = false;
        if (amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawal of ₹ " + amount + "
successful.");
       } else {
            System.out.println("Insufficient funds. Withdrawal not
allowed.");
```

```
        }
    }

    public void displayBalance() {
          update();
        System.out.println("Account Balance: ₹ " + balance);
    }

    public abstract void update();
}


class CurAcct extends Account {
    double minBalance;
    double serviceCharge;

    public CurAcct(String customerName, long accountNumber, double
balance) {
        super(customerName, accountNumber, "Current", balance);
        this.minBalance = 500;
        this.serviceCharge = 10;
    }

    public void update() {
        if(!updated) {
        if (balance < minBalance) {
            balance -= serviceCharge;
            System.out.println("Service charge of ₹ " + serviceCharge
+ " imposed for falling below minimum balance.");
        }
        updated = true;
        }
    }
}
```

```
class SavAcct extends Account {
   double interestRate;

   public SavAcct(String customerName, long accountNumber, double
balance) {
        super(customerName, accountNumber, "Savings", balance);
        this.interestRate = 0.05;
   }

   public void update() {
        double interest = balance * interestRate;
        balance += interest;
        System.out.println("Interest of ₹ " + interest + " computed
and deposited. Updated balance: ₹ " + balance);
   }
}

public class BankDemo {
   public static void main(String[] args) {
        CurAcct currentAccount = new CurAcct("Priyanshu", 123456789,
1000);
        SavAcct savingsAccount = new SavAcct("Ajit", 987654321, 5000);

        currentAccount.deposit(200);
        currentAccount.displayBalance();
        currentAccount.withdraw(200);
        currentAccount.displayBalance();

        savingsAccount.deposit(1000);
        savingsAccount.displayBalance();
        savingsAccount.withdraw(800);
        savingsAccount.displayBalance();
   }
}
```

Lab 6

```
public class Program1 {
   public static void main(String[] args) {

        System.out.println("Using String Literal");
        String str1 = "Hello, World!";
        System.out.println("String 1: " + str1);

        System.out.println("Using new keyword");
        String str2 = new String("Apple");
        System.out.println("String 2: " + str2);

        System.out.println("Using char array");
        char[] charArray = {'O', 'r', 'a', 'n', 'g', 'e'};
        String str3 = new String(charArray);
        System.out.println("String 3: " + str3);

        System.out.println("Using byte array with character
encoding");
        byte[] byteArray = {72, 101, 108, 108, 111}; // ASCII values
for "Hello"
        String str4 = new String(byteArray);
        System.out.println("String 4: " + str4);

        System.out.println("Substring of an existing string");
        String originalStr = "Java Programming";
        String str5 = new String(originalStr.substring(0, 4)); //
Extract "Java"
        System.out.println("String 5: " + str5);

        System.out.println("Using StringBuilder");
        StringBuilder stringBuilder = new StringBuilder("Hello");
        String str6 = new String(stringBuilder.toString());
```

```
        System.out.println("String 6: " + str6);

    }
}


public class Program234 {
    public static void main(String[] args) {

        String str1 = "Hello";
        String str2 = "World";
        int length = str1.length();

        System.out.println("Program 2");
        System.out.println("String 1: " + str1);
        System.out.println("String 2: " + str2);
        System.out.println("Length of String 1: " + length);
        String concatenatedString = str1.concat(", " + str2);
        System.out.println("Concatenated String: " +
concatenatedString);

          System.out.println("Program 3");
        String toStringExample = "This is an example.";
        System.out.println("Using toString(): " + toStringExample);

          System.out.println("Program 4");
        String originalString = "Welcome to Bmsce college";
        char[] targetArray = new char[5];
        originalString.getChars(11, 16, targetArray, 0);
        String extractedString = new String(targetArray);
        System.out.println("Extracted String: " + extractedString);

    }
}
```

```
public class Program5 {
   public static void main(String args[])
   {
       String str = "Abracadbara";
         byte[] ba = str.getBytes();
       char[] ca = str.toCharArray();

         System.out.println("Bytes:");
       for (int i = 0; i < ba.length; i++) {
           System.out.println(ba[i]);
       }

       System.out.println("\nCharacter Array:");
       for (int i = 0; i < ca.length; i++) {
           System.out.println(ca[i]);
       }
   }
}


public class Program6 {
   public static void main(String[] args)
       String str1 = "Bmsce";
       String str2 = "College";
       String str3 = "BMSCE";
        System.out.println("Bmsce equals Bmsce -> " +
str1.equals(str1));
         System.out.println("Bmsce equals College -> " +
str1.equals(str2));
         System.out.println("Bmsce equals BMSCE -> " +
str1.equals(str3));
         System.out.println("Bmsce equalsIgnoreCase BMSCE -> " +
str1.equalsIgnoreCase(str3));
   }
```

```
}


public class Program7 {
   public static void main(String[] args) {
        System.out.println("Program 7");
        String str1 = "Welcome to Bmsce College of Engineering";
        String str2 = "Bmsce college";

        if (str1.regionMatches(true, 11, str1, 0, str2.length())
            System.out.println("Substring is matched");
        } else {
            System.out.println("Substring is not matched");
        }
   }
}


public class Program8910 {
   public static void main(String[] args) {
     System.out.println("Program 8");
     System.out.println("Mango startsWith Man -> " +
"Mango".startsWith("Man"));
   System.out.println("Program 9");
     System.out.println("Orange endsWith range -> " +
"Orange".startsWith("range"));
     System.out.println("Program 10");
     String str1 = "Apple";
     String str2 = new String("Apple");
     System.out.println("String 1: " + str1);
     System.out.println("String 2: " + str2);
     System.out.println("String 1 equals String 2 -> " +
str1.equals(str2));
     System.out.println("String 1 == String 2 -> " + (str1 == str2));
   }
```

```
}
```

Lab 7

```
// CIE/Internals.java

package CIE;

import java.util.Scanner;

public class Internals extends Student {
   protected int marks[] = new int[5];

   public void inputCIEMarks() {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter CIE Marks:");
        for (int i = 0; i < 5; i++) {
            System.out.println("Enter marks for Course: " + (i+1) +
":");
            marks[i] = sc.nextInt();
        }
        sc.close();
   }
}

// CIE/Student.java
package CIE;

import java.util.Scanner;

public class Student {
   protected String usn = new String();
   protected String name = new String();
   protected int sem;

   public void inputStudentDetails() {
```

```
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter your usn: ");
        usn = sc.nextLine();
        System.out.print("Enter your name: ");
        name = sc.nextLine();
        System.out.print("Enter your semester: ");
        sem = sc.nextInt();
        sc.close();
    }

    public void displayStudentDetails() {
        System.out.println("USN: " + usn);
        System.out.println("Name: " + name);
        System.out.println("Semester: " + sem);
    }
}


// SEE/Externals.java
package SEE;

import CIE.Internals;
import java.util.Scanner;

public class Externals extends Internals {
    protected int marks[];
    protected int finalMarks[];

    public Externals() {
        marks = new int[5];
        finalMarks = new int[5];
    }

    public void inputSEEMarks() {
        Scanner sc = new Scanner(System.in);
```

```
        System.out.println("Enter SEE marks:");
        for (int i = 0; i < 5; i++) {
            System.out.println("Course " + (i + 1) + ":");
            marks[i] = sc.nextInt();
        }
        sc.close();
    }

    public void calculateFinalMarks() {
      for (int i = 0; i < 5; i++) {
            finalMarks[i] = (marks[i] / 2) + super.marks[i];
        }
    }

    public void displayFinalMarks() {
        displayStudentDetails();
        for (int i = 0; i < 5; i++) {
            System.out.println("Course " + (i + 1) + ":" +
finalMarks[i]);
        }
    }
}

// Main.java
import SEE.Externals;

public class Main {
    public static void main(String[] args) {
        int numOfStudents = 2;
        External[] finalMarks = new External[numOfStudents];

        for (int i = 0; i < numOfStudents; i++) {
            finalMarks[i] = new External();
            finalMarks[i].inputStudentDetails();
            finalMarks[i].inputCIEMarks();
```

```
            finalMarks[i].inputSEEMarks();
        }

        System.out.println("Displaying data:");
        for (int i = 0; i < numOfStudents; i++) {
            finalMarks[i].calculateFinalMarks();
            finalMarks[i].displayFinalMarks();
        }
    }
}
```

Lab 8

```
import java.util.Scanner;

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

class Father extends InputScanner {
   private int fatherAge;

   public Father() throws WrongAge {
       super();
       super.s = new Scanner(System.in);
       System.out.print("Enter father's age: ");
       fatherAge = super.s.nextInt();
       super.s.close();
       if (fatherAge < 0) {
           throw new WrongAge("Age cannot be negative");
       }
   }

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

class Son extends Father {
   private int sonAge;

   public Son() throws WrongAge {
       super();
```

```
        super.s = new Scanner(System.in);
        System.out.print("Enter son's age: ");
        sonAge = super.s.nextInt();
        super.s.close();
        if (sonAge >= super.fatherAge) {
            throw new WrongAge("Son's age cannot be greater than or
equal to father's age");
        } else if (sonAge < 0) {
            throw new WrongAge("Age cannot be negative");
        }
    }

    public void display() {
        super.display();
        System.out.println("Son's age: " + sonAge);
    }
}

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

class InputScanner {
    Scanner s;
    public InputScanner() {
    }
}
```

Lab 9

```
class CollegeThread extends Thread {
    @Override
    public void run() {
        while (true) {
            System.out.println("BMS College of Engineering");
            try {
                Thread.sleep(10000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }}}
}

class DepartmentThread extends Thread {
    @Override
    public void run() {
        while (true) {
            System.out.println("CSE");
            try {
                Thread.sleep(2000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }))
}

public class Main {
    public static void main(String[] args) {
        CollegeThread collegeThread = new CollegeThread();
        DepartmentThread departmentThread = new DepartmentThread();
        collegeThread.start();
        departmentThread.start();
    }
}
```

Lab 10

```
// Inter-Process Communication

class Q {
   private int n;
   private boolean valueSet = false;

   synchronized int get() {
       while (!valueSet) {
           try {
               System.out.println("\nConsumer waiting\n");
               wait();
           } catch (InterruptedException e) {
               System.out.println("InterruptedException caught");
           }
       }

       System.out.println("Got: " + n);
       valueSet = false;
       System.out.println("\nNotify Producer\n");
       notify();
       return n;
   }

   synchronized void put(int n) {
       while (valueSet) {
           try {
               System.out.println("\nProducer waiting\n");
               wait();
           } catch (InterruptedException e) {
               System.out.println("InterruptedException caught");
           }
       }
```

```
        this.n = n;
        valueSet = true;
        System.out.println("Put: " + n);
        System.out.println("\nNotify Consumer\n");
        notify();
    }
}

class Producer implements Runnable {
    private Q q;

    Producer(Q q) {
        this.q = q;
        new Thread(this, "Producer").start();
    }

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

class Consumer implements Runnable {
    private 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++;
        }
    }
}

public class PC {
    public static void main(String args[]) {
        Q q = new Q();
        new Producer(q);
        new Consumer(q);
        System.out.println("Press Control-C to stop.");
    }
}

// Deadlock
public class DeadlockExample {
    public static void main(String[] args) {
        Object lock1 = new Object();
        Object lock2 = new Object();

        Thread thread1 = new Thread(() -> {
            synchronized (lock1) {
                System.out.println("Thread 1 acquired lock1");

                try {
                    Thread.sleep(100);
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }

                synchronized (lock2) {
                    System.out.println("Thread 1 acquired lock2");
```

```
                }
            }
        });

        Thread thread2 = new Thread(() -> {
            synchronized (lock2) {
                System.out.println("Thread 2 acquired lock2");

                try {
                    Thread.sleep(100);
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }

                synchronized (lock1) {
                    System.out.println("Thread 2 acquired lock1");
                }
            }
        });

        thread1.start();
        thread2.start();
    }
}
```

Lab 11

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

class SwingDemo{
   SwingDemo(){
        JFrame jfrm = new JFrame("Divider App");
        jfrm.setSize(275, 150);
        jfrm.setLayout(new FlowLayout());
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JLabel jlab = new JLabel("Enter the divider and dividend:");

        JTextField ajtf = new JTextField(8);
        JTextField bjtf = new JTextField(8);

        JButton button = new JButton("Calculate");

        JLabel err = new JLabel();
        JLabel alab = new JLabel();
        JLabel blab = new JLabel();
        JLabel anslab = new JLabel();

        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 l = new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        System.out.println("Action event from a text field");
    }
};
ajtf.addActionListener(l);
bjtf.addActionListener(l);

button.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        try{
            int a = Integer.parseInt(ajtf.getText());
            int b = Integer.parseInt(bjtf.getText());
            int ans = a/b;

            alab.setText("\nA = " + a);
            blab.setText("\nB = " + b);
            anslab.setText("\nAns = "+ ans);
        }
        catch(NumberFormatException e){
            alab.setText("");
            blab.setText("");
            anslab.setText("");
            err.setText("Enter Only Integers!");
        }
        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();
            }
        });
    }
}
```