

B.M.S COLLEGE OF ENGINEERING BENGALURU
Autonomous Institute, Affiliated to VTU



OBJECT ORIENTED JAVA PROGRAMMING

Submitted by:

**Name : Shreyansh Sethiya
1BM22CS269**

Faculty incharge:

Dr Seema Patil

Department of Computer Science and Engineering
B.M.S College of Engineering
Bull Temple Road, Basavanagudi, Bangalore 560 019

Name → Shreyash Sathija

USN → 18M2205269

INDEX

| SI No. | DATE | title |
|-----------|----------|---------------------|
| 1) | 12/12/23 | Lab Program 1 |
| 2) | 19/12/23 | Lab Program 2 |
| 3) | 26/12/23 | Lab Program 3 |
| 4) | 2/1/24 | Lab Program 4 |
| 5) | 16/1/24 | Lab Program 5 |
| 6) | 23/1/24 | Lab Program 6 |
| 7) | 30/1/24 | Lab Program 7 |
| 8) | 6/2/24 | Lab Program 8 |
| 9) | 13/2/24 | Lab Program 10 a, b |
| 10 | 20/2/24 | Lab Program 9 |

LAB Program 2

Roots of a Quadratic Equation

12/12/2022

```
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();
```

```
}
```

$$d = b^2 - 4ac$$

```
y (d==0)
```

```
{
```

$$r_1 = (-b) / (c \sqrt{a})$$

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

```
System.out.println("Root 1 = Root 2 = " + r1);
```

```
}
```

Name : Shreyansh Sethia

USN → 10M 22 CS 269

out

else if (cd > 0)

{

$$\gamma_2 = ((-b) + (\text{Math.sqrt(cd)})) / (\text{double}(2 * a));$$

$$\delta_2 = ((-b) - (\text{Math.sqrt(cd)})) / (\text{double}(2 * a));$$

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

System.out.println("Root1 = " + \gamma_2 + " Root2 = " + \delta_2);

}

else if (cd < 0)

{

System.out.println("Roots are imaginary");

$$\gamma_2 = (-b) / (2 * a);$$

$$\delta_2 = \text{Math.sqrt}(-d) / (2 * a);$$

System.out.println("Root1 = " + \gamma_2 + " + i " + \delta_2);

System.out.println("Root2 = " + \gamma_2 + " - i " + \delta_2);

}

3

3

class QuadraticMain

{

public static void main(String args[])

{

Quadratic = new Quadratic();

Q.getd();

Q.compute();

System.out.println("Answers are " + Q.getans() + " " + Q.getans2());

3

3

output \rightarrow

1) Enter the coefficients of a, b and c

4

-4

1

Roots are real and equal

Root 1 = Root 2 = 1/2

Shreyansh Sutroja 18M22C5269

2) Enter the coefficient of a, b and c

2

-9

10

Roots are real and distinct

Root 1 = 7.7 Root 2 = 1.2

Shreyansh Sutroja 18M22C5269

3) Enter the coefficients of a, b & c

0

0.2

2

Not a quadratic equation

Enter a non-zero value for a

1

Roots are imaginary

Root 1 = -12 + i(8.4852)

Root 2 = -12 - i(8.4852)

Shreyansh Sutroja 18M22C5269

16

Jan
12/12/23

05
11/3

C++ Programs 2

- Q. Develop a java program to create a class student with marks, name, name, on array credits and on 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
```

```
{  
    Subject subject[8];  
    String name;  
    String usn;  
    double sgpa = 0;  
    Scanner s;
```

```
Student()  
{  
    int i;  
    Subject = new Subject[8];  
    for(i=0; i<8; i++)  
        subject[i] = new Subject();  
    s = new Scanner(System.in);  
}
```

```
void getStudentDetails()  
{  
    System.out.println("Enter your name:");  
    name = s.nextLine();  
    System.out.println("Enter your usn:");  
    usn = s.nextLine();  
}
```

void getMarks()

{
for (int i=0; i<8; i++)

{

System.out.println("Enter marks for subject " + i + ":");
subject[i].subjectMarks = s.nextInt();

System.out.println("Enter credits for subject " + i + ":");
subject[i].credits = s.nextInt();

subject[i].grade = (subject[i].subjectMarks / 10) + 2;

if (subject[i].grade > 10)

subject[i].grade = 10;

if (subject[i].grade <= 4)

subject[i].grade = 0;

}

}

void computeSGPA()

{

int totalCredits = 0;

for (int i=0; i<8; i++)

{

Sgpa += (subject[i].credits * subject[i].grade);

totalCredits += subject[i].credits;

}

sgpa = sgpa / totalCredits;

}

}

Class.main

{

public static void main (String args[])

{

Student s2 = new Student();

s2.getStudentDetails();

s2.getMarks();

s2.computeSGPA();

System.out.println("Name" + s2.name);
System.out.println("Usn" + s2.usn);
System.out.println("SPPA" + s2.sppa);

7

8

Output

Enter your Name:

Satyendra

Enter your USN:

26mcse269

Enter marks for subject 1:

90

credits

Enter marks for subject 2:

4

Enter marks for subject 2:

95

Enter credits for subject 2:

4

Enter marks for subject 3:

80

Enter credits for subject 3:

63

Enter marks for subject 4:

85

Enter credits for subject 4:

3

Enter marks for subject 5:

78

Enter credits for subject 5:

2

Enter credits for subject 6:

79

Enter credit for subject 6:

1

Enter marks for subject 7:

60

Enter credit for subject 7:

1

Enter marks for subject 8:

99

Enter credit for subject 8:

1

Name : Shreyansh

UIN : 26m22-5269

SYRN : 9-210526

Fri 19/2/23

⑩

CAR 3

import java.util.Scanner;

class Books

{

String name; String author;

int price; int numPages;

Books (String name, String author, int price, int numPages)

this.name = name;

this.author = author;

this.price = price;

this.numPages = numPages;

}

public String toString() {

String name, author, price, numPages;

name = "Book name" + this.name + "\n";

author = "Author name" + this.author + "\n";

price = "Price:" + this.price + "\n";

numPages = "Number of Pages:" + this.numPages + "\n";

return name + author + price + numPages;

}

Class Main

{

```
public static void main (String args [] )
```

```
{
```

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

```
int n, price, numPages;
```

```
String name, author;
```

```
n = s.nextInt();
```

```
Books b [] ;
```

```
b = new Books [n];
```

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

```
name = s.next();
```

```
author = s.next();
```

~~price = s.nextInt();~~~~numPages = s.nextInt();~~

```
b [i] = new Books (name, author, price, numPages);
```

```
}
```

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

```
{
```

```
System.out.println (b [i].toString());
```

```
}
```

```
System.out.println ("Shreyash Sutija 18M22C52X9");
```

"); output → 2

Rich Dad Poor Dad

Richard

600

250

The Post Office
Rosedale Notts Toton

300

650

Book Name: Rich Dad Poor Dad

Author Name: Richard

Price : 6.00

Number of Pages 256

Book Name: The Post Office

Author Name: Rabindra Nath Tagore

Price : 3.00

Number of Pages : 650

26/12/23

- Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea(). Provide three classes extending the class Shape: Rectangle, Circle and Square such that each one of the classes contains only the method printArea() that prints the area of the given shape.

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

```
class InputScanner {

```

```
    Scanner s;
```

```
InputScanner() {

```

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

```

y



y

```
abstract class Shape extends InputScanner {

```

```
    double a;

```

```
    double b;

```

```
    abstract void getArea();

```

```
    abstract void displayArea();

```

y

```
class Rectangle extends Shape {

```

```
    void getArea() {

```

System.out.println("Finding the area of rectangle");

System.out.println("Enter the length of rectangle");

a = s.nextDouble();

System.out.println("Enter the width of rectangle");

b = s.nextDouble();

y

```
void displayArea() {
```

```
System.out.println("The area of rectangle is " + abc);
```

y

y

```
class Rectangle extends Shape {
```

```
void getInputs() {
```

```
System.out.println("Enter the area of triangle");
```

```
System.out.println("Enter the height of triangle");
```

```
a = s.nextInt();
```

```
System.out.println("Enter the base of triangle");
```

```
b = s.nextInt();
```

y

```
void displayArea() {
```

```
System.out.println("The area of triangle is " + 0.5 * a * b);
```

y

y

```
class Circle extends Shape {
```

```
void getInputs() {
```

```
System.out.println("Finding the area of circle");
```

```
System.out.println("Enter the radius of circle");
```

```
a = s.nextInt();
```

y

```
void displayArea() {
```

```
System.out.println("The area of circle is " + 3.14f * a * a);
```

y

y

```

public class AbstractClassEx2
{
    public static void main (String[] args)
    {
        Rectangle d = new Rectangle ();
        d.getInputs();
        d.displayArea();
    }

    Triangle t = new Triangle ();
    t.getInputs();
    t.displayArea();

    Circle c = new Circle ();
    c.getInputs();
    c.displayArea();
}

```

System.out.println ("Area of rectangle
200*250=50000");

Output → Finding the area of rectangle
Enter the length of rectangle

10

Enter the width of rectangle

10

The area of rectangle is 100.0

Finding the area of triangle

Enter the height of triangle

10

Enter the base of triangle

10

The area of triangle is 50.0

Finding the area of circle

Enter the radius of circle

10

The area of circle is 314.0000104908175

AB 02/04/24 Shreyash Sitriga I BM 22 CS 269

Bank

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

```
class Account {
```

```
    String customerName;
```

```
    int accountNumber;
```

```
    String accountType;
```

```
    float balance;
```

```
    Account (String customerName, int accountNumber,  
             String accountType);
```

```
    } this.customerName = customerName;
```

```
    this.accountNumber = accountNumber;
```

```
    this.accountType = accountType;
```

```
    this.balance = 0.0;
```

```
}
```

```
public void displayBalance () {
```

```
    System.out.println ("Account Balance $" + bal-
```

```
}
```

```
void withdraw (float amount) {
```

```
    if (balance >= amount) {
```

```
        balance = balance - amount;
```

```
        System.out.println ("withdrawal of $" + amount +
```

```
"successful") ;
```

```
else {
```

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

```
}
```

```
}
```

```
}
```

class BankAccount extends Account {
float interest;
float interestRate;
public BankAccount(String customerName, int accountNumber)
{
super(customerName, accountNumber, "savings");
this.interestRate = 0.05;
}
}

7
void computeInterest() {
double interest = balance * interestRate;
deposit(interest);
System.out.println("Interest of \$" + interest + " deposited");
}

8
void deposit(float amount) {
balance = balance + amount;
System.out.println("Deposit of \$" + amount + " successful");
}

9
class Current extends Account {
float minBalance;
float serviceCharge;
public Current(String customerName, int accountNumber)
{
super(customerName, accountNumber, "current");
this.minBalance = 1000.0;
this.serviceCharge = 50.0;
}
}

10
public void displayBalance() {
super.displayBalance();
}

if a balance < minbalance) {

System.out.println ("service charge of 5" + servicecharge
"imposed due to low balance");

balance = balance - servicecharge;

System.out.println ("Totally charged ");

Output :-

public class Bank {

public static void main (String [] args) {

Scanner scanner = new Scanner (System.in);

Savaccet savingsAccount = new Savaccet ("Ahu", 1000);

Current currentAccount = new Current ("Kol", 2000);

Savings Account deposit (500);

Savings Account . computeInterest ();

Savings Account . display Balance ();

System.out.println ("Enter amount to withdraw

from current account");

float withdrawAmount = scanner . nextFloat ();

Current Account . withdraw (withdrawAmount);

Current Account . display Balance ();

System.out.println ("Enter amount to withdraw

from savings account");

float withdrawAmountSavings = scanner . nextFloat ();

Savings Account . withdraw (withdrawAmountSavings);

Savings Account . display Balance ();

Scanner . close ();

Output →

500 is deposited successfully.

Balance is 525

Enter amount to withdraw from the ~~saving~~ current account

125

Balance → 400

creencias →

use & java program to create a generic class
Stack which holds 5 integers and 5 doubles.
import java.util.*;

class Stack {

 Stack();

 int top;

 int size = 10;

 Stack() {

 size = (1+1) new Object [size];

 top = -1;

 }

 void push (item) {

 if (top == size - 1) {

 System.out.println("Overflow");

 } else {

 else {

 stack [++top] = item;

 }

 }

 }

 T pop() {

 if (top < 0) {

 System.out.println("Underflow");

 return null;

 }

 else {

 return stack [top - 1];

 }

 }

 }

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

 stack < Integer > S1 = new Stack<Integer>;
 stack < Double > S2 = new Stack<Double>;

 Scanner S = new Scanner (System.in);
 System.out.println ("Enter elements in
 integer stack");

 }

}

for (int i=0; i<5; i++) {
 int n = S.nextInt();
 S1.push(n);
}

}

System.out.println ("Enter elements in
double stack");

}

for (int i=0; i<5; i++) {
 double m = S.nextDouble();
 S2.push(m);
}

}

System.out.println ("Elements of S2");

}

for (int i=0; i<5; i++) {
 System.out.println (S2.pop());
}

}

S.clear();

}

}

output → Enter elements in integer stack

1 3 5 9 12

Enter elements in double stack

4 5 7 8

Elements in S1

12
9
6
3
1

Elements in S2

8.0
7.0
5.0
4.0
2.0

white a java program to create a abstract class Shape
abstract methods calculateArea() and calculatePerimeter()
create subclasses Circle and Triangle that extends the shape
class.

abstract class Shape {

public abstract double calculateArea();

public abstract double calculatePerimeter();

3

class Circle extends Shape {

private double radius;

public Circle (double radius) {

this.radius = radius;

4

public double calculateArea () {

return Math.PI * radius * radius;

5

public double calculatePerimeter () {

return 2 * Math.PI * radius;

6

7

class Triangle (double side1, double side2, double side3)

8

this.side1 = side1;

this.side2 = side2;

this.side3 = side3;

9

public

return

3

public

3

public

Tr

Y

Output

Re

public class calculateArea {

double s = (side1 + side2 + side3) / 2;

return Math.sqrt(s * (s - side1) * (s - side2) *

(s - side3));

public double calculatePerimeter () {

return side1 + side2 + side3;

public class Main {

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

Circle circle = new Circle (5.0);

System.out.println (" Circle-Area: " + circle.calculateArea ());

" Circle-Perimeter: " + circle.calculatePerimeter ());

Triangle triangle = new Triangle (3.0, 4.0, 5.0);

System.out.println (" Triangle-Area: " + triangle.calculateArea ());

+ " Perimeter: " + triangle.calculatePerimeter ());

Output → Circle-Area: 78.5398 Perimeter: 31.415

Triangle-Area: 6.0 Perimeter: 12.0

X Write a Java program to create a generic class which holds 5 integers and 5 double values.

Demonstrated various string constructor with help of Java programs

```
class String  
{  
    public String void main (String args [] )  
    {  
        char [] c = {'J', 'a', 'v', 'a'};  
        String s2 = newString (c);  
        System.out.println (s2);  
        System.out.println (s2);  
    }  
}
```

Output → JAVA
JAVA

Demonstrated startWith () to give output true/false

public class Main

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

String test = "teststring";

String pattern = "te";

System.out.println (test.startsWith (pattern));

Pattern = "est";

System.out.println (test.startsWith (pattern));

8/10/24

Output →
true
false

CAT-6

I wrote a package CIE which has two classes - student and InternMarks. The class InternMarks derived from Student has an array that stores the internal marks score in five courses of the current semester & the student. I wrote another package SEM which has the class InternMark which is derived class of student. The class has an array that stores marks score in five courses of the current semester.

public class Student {

 public String name;

 public String marks;

 public int sem;

 public Student (String name, String marks, int sem) {

 this.name = name;

 this.marks = marks;

 this.sem = sem;

 }

 }

 public class InternMarks extends Student {

 public int[] marks;

 public InternMarks (String name, String marks, int sem) {

 super.name, marks, sem);

 }

 public void inputCIEmarks () {

 Scanner s = new Scanner (System.in);

 for (int i = 1; i <= 5; i++) {

 System.out.println ("Enter marks for " + i + " subject");

 marks[i] = s.nextInt();

```
package SEE;
import CSE.External;
import java.util.Scanner;
public class External extends Internals {
    public int marks[] {
        protected int finalMarks[];
    }
    public External() {
        marks = new int[5];
        finalMarks = new int[5];
    }
    public void inputSEEmarks() {
        Scanner s = new Scanner(System.in);
        for (int i=0; i<5; i++) {
            System.out.print("Subject " + (i+1) + " marks: ");
            marks[i] = s.nextInt();
        }
    }
    public void calculateFinalMarks() {
        for (int i=0; i<5; i++) {
            finalMarks[i] = marks[i]/2 + Super.marks;
        }
    }
    public void displayFinalMarks() {
        displayStudent();
    }
}
```

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

System.out.println ("Subject" + (i+1) + ":" + findMarks());

Y
Main.java →

import SEE.externals;

public class Main {

public static void main (String args[])

{

int num of students = 2;

External findMarks() = new External()

[num of students];

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

findMarks[i] = new External();

findMarks[i].input StudentDetails();

System.out.println ("Enter CIE marks");

findMarks[i].input CIEMarks();

System.out.println ("Enter SEE marks");

findMarks[i].input SEE_Marks();

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

for (int i = 0; i < num of Students; i++) {

findMarks[i].calulateFinalMarks();

findMarks[i].displayFinalMarks();

Y
Y

Y
Y

Y
Y

Output →

Enter USN : 18M22CS273

Enter Name : Shreyansh Sittya

Enter Semester : 2

Enter Internal Marks for Shreyansh

Subject 1 Marks : 40

Subject 2 Marks : 35

Subject 3 Marks : 42

Subject 4 Marks : 38

Subject 5 Marks : 45

Enter SEE marks for Shreyansh

Subject 1 Marks : 75

Subject 2 Marks : 80

Subject 3 Marks : 65

Subject 4 Marks : 70

Subject 5 Marks : 85

Enter USN : 18M22CS269

Enter USN : Shreyansh

Enter Semester : 2

Enter CIE Marks

Enter Internal Marks for Shreyansh

Subject 1 : 36

Subject 2 : 48

Subject 3 : 50

Subject 4 : 42

Subject 5 : 55

Enter SEE Marks for Shreyansh

Subject 1 Marks : 80

Subject 2 Marks : 35

Subject 3 Marks : 42

Subject 4 Marks : 38

Subject 5 Marks : 45

Enter ^{Enter} SEE marks for Shreyas

Subject 1 Marks 75

Subject 2 Marks 80

Subject 3 Marks 42

Subject 4 Marks 38

Subject 5 Marks 45

Enter SEE marks for Shreyas

Subject 1 Marks : 80

Subject 2 Marks : 72

Subject 3 Marks : 85

Subject 4 Marks : 78

Subject 5 Marks : 90

ii Display Data

URN = 10122CS209

Name = Shreyas

Semester = 2

Subject 1 77

Subject 2 75

Subject 3 79

Subject 4 73

Subject 5 87

URN = 10122CS209

Name = Shreyas

Semester = 2

Subject 1 = 77

Subject 2 = 84

Subject 3 = 92

Subject 4 = 82

Subject 5 = 100

Shreyas
30.01.24

CAB-7

* WAP of inheritance tree if son's age >= father's age.

import java.util.Scanner;

class wrongAge extends exception {

 public wrongAge(String s) {

 System.out.println(s);

 }

class Father {

 int age;

 public Father(int age) {

 this.age = age;

 }

 public void checkAge() throws wrongAge {

 if (age < 0)

 {

 throw new wrongAge("Age cannot
 be negative");

 }

 }

 public void display() {

 System.out.println("Father's age = " + age);

 }

class Son extends Father {

 public Son(int age) {

 super(age);

 }

int sage = age;

public void checkAge() throws WrongAgeException
if (sage < 0)
2

throws new WrongAge("Age cannot
be negative");

3

else if (sage >)

2

throws new WrongAge("Son cannot
older than father");

else if (sage == age)

2

throws new WrongAge("Son cannot
be same as Father");

3

4

public void display()

System.out.println ("Son's age = " + sage);

5

public class AgeException {

public static void main (String[] args) {

Scanner sc = new Scanner (System.in);

System.out.println ("Enter age of father");

Father f = new Father (sc.nextInt());

System.out.println ("Enter age of son");

Son's age = new son's current age - 7;

Sc. class(); // displays user input

S. age = F. age;

try &

f. change();

S. change();

S. display();

d. display();

y

catch (Wrong Age e)

e

System.out.println(e);

y

y

y

Output →

Enter age of Father

50

Enter the age of son

-10

Age cannot be negative

Wrong Age

Java Exception Age

Enter age of Father

20

Enter age of son

30

Son cannot be older than Father

Wrong Age

Java Exception Age

Enter age of Father

20

Enter age of son

20

Son cannot be same age as Father
wrong age

java Encapsulation Age

Enter age of Father

30

Enter age of son

5

Son's Age = 5

Father's Age = 30

30 is maximum age of father
about 5 must be minimum age

- O Write a program which creates two threads one thread displays "BMS college of engineering" & every 10 second another displaying "EST" on every 2nd

M
class Message implements Runnable {

String message;

int interval;

public Message(String message, int interval)

L this.message = message;

this.interval = interval;

}

public void run() {

try {

while (true) {

System.out.println(message);

Thread.sleep(interval);

}

}

catch (InterruptedException e) {

System.out.println("Thread Interrupted");

}

}

}

public class Est {

public static void main(String args) {

L

Thread thread2 = new Thread

(new Message("BMS college of
Engineering", 10000));

thread2.start();

Thread thread2 = new Thread (new Runnable() {
 public void run() {
 System.out.println("Hello from thread2");
 }
});

x

y

Output → BMS college of Engineering

CSE
CSW
CSC
CSE
BMS

Q Write a program that creates a user interface
to perform integer divisions. The user enters two numbers
in the text fields, num₁ and num₂. The division
of num₁ and num₂ is displayed in the result
field when the Divide button is clicked. If num₁
or num₂ was not an integer, the program
would throw a NumberFormat Exception. If num₂
was zero, the program would throw an Arithmetic
Exception. Display the exception in a message dialog.

```
import javax.swing.*;  
import java.awt.*;  
import java.awt.event.*;  
  
class SwingDemo extends JFrame  
{  
    // create JFrame container  
    JFrame frm = new JFrame("Divide app");  
    frm.setSize(275, 150);  
    frm.setLayout(new FlowLayout());  
    frm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
  
    JPanel lab = new JPanel("Enter the divisor and dividend");  
    JTextField nf1 = new JTextField(8);  
    JTextField nf2 = new JTextField(8);  
  
    JButton button = new JButton("Calculate");  
    JPanel err = new JPanel();  
    JPanel ans = new JPanel();  
  
    JPanel res = new JPanel();
```

JLabel onsel = new JLabel();

jfrm.add(ers);

jfrm.add(fld);

jfrm.add(cjty);

jfrm.add(cbyt);

jfrm.add(button);

jfrm.add(cold);

jfrm.add(csls);

jfrm.add(cnsld);

ActionListener I = new ActionListener() {

public void actionPerformed(ActionEvent evt) {

System.out.println("Action event from a
tent field");

3

3; 3 (done with some hour later)

ajy.addActionListener();

bjf.addActionListener();

button.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent evt) {

try {

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

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

int ans = a/b;

ans.setText("m A = " + a);

ans.setText("m X = " + b);

ans.setText("m Ans = " + ans);

3

cotab & Number Formed Enclosures (7) 5

old-set tent C " 171

old. set tent c° "75

```
anslos.setTemp(" "));
```

corr. set tent ("Enter only *Tortuga*,");

三

cates *C. Antennatus* Enoplos & L

old · retent (" ");

lets. Set tent ("");

and set tent ("");

crr. set Int c" I should be now ready");

3

۲۷

from . setVisible (true);

3

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

Swing Utilities invoke later (now Runnable) &

public road surc 2

more swing down);

2

7

2

Output →

Divide A/B - \Rightarrow X
Enter the dividend and divisor

(b) Divide APP -D X

Enter the divisor and dividend

| | |
|----|---|
| 10 | 0 |
|----|---|

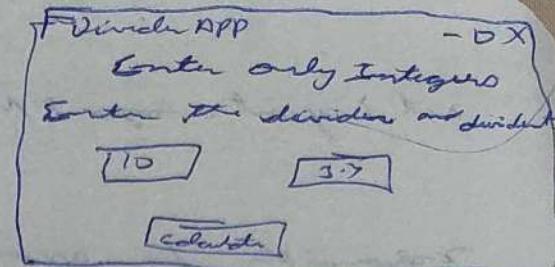
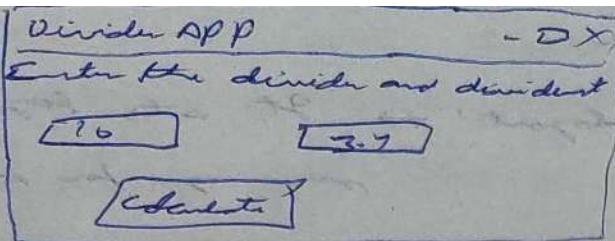
Calculator

Binder APP + -D X
B should not exceed
Enter the divisor and dividend

| | |
|----|---|
| 10 | 0 |
|----|---|

(decimal)

Q4



Report on AWT functions \Rightarrow

JFrame \rightarrow Represent main window of application.

JLabel \rightarrow used to display text labels or text.

add() \rightarrow This is used to add a swing component
 - button label text field \rightarrow to a container
 (JFrame).

setText(text) \rightarrow It is used to set the text of text
 based component (such as label) to
 specified string.

getText() \rightarrow get others text from text based component
 get represent string representing current
 text.

addActionListener < ActionListener interface > \rightarrow This method
 adds.

action listener to the component that generates action
 events (Jbutton) . when action occurs action
 performed() method invoked

setSize (int height int width) \rightarrow It is used to set
 size of component
 such as JFrame to specified height and width in
 pixels.

set layout alignment Margin layout) \rightarrow It sets layout margins for container responsible for arranging elements inside it.

set visibility & position visible) \rightarrow It sets visibility of component when other component and concert becomes visible or may and hidden were set to false.

Sh
20.02.24

CAB-10
O Demonstrate intercommunication and deadlock

class O2

int m;

boolean valueSet = false;

synchronized int get <= t

while c, valueSet)

try {

System.out.println ("In consumer Waiting(m);
wait();

}

catch InterruptedException e) {

System.out.println ("Interrupted Exception");

}

System.out.println ("not:" + m);

value set = false;

System.out.println ("Intermediate Product " + m);

notify();

return m;

}

synchronized void put (int n) {

while c, value set)

try {

System.out.println ("Product " + m);

wait();

}

catch (InterruptedException)

{

System.out.println ("Interrupted Exception
caught");

}

this.n = n;
valueSet = true;
System.out.println("Put: " + n);
System.out.println("Initial consumer");
notifyAll();

3

3

class Producer implements Runnable {

② 2;

Producer (Q2) {

this.q = q;

new Thread(this, "producer").

start();

3

public void run () {

int i = 0;

while (i < 15) {

q.put (i++);

3
3

3

class Consumer implements Runnable {

② 2;

Consumer (Q2) {

this.q = q;

new Thread(this, "consumer").

start();

3

public void run () {

int i = 0;

while (i < 15) {

int r = q.get ();

System.out.println("consumed : " + o);
++i;

30

30

30

class PCEnd {

public static void main (String args[]) {

2

o q = new o;

new Producer (q);

new consumer (q);

30

30

Output:

Put: 0

Intimate consumer

producer waiting

creat: 0

Intimate Producer

consumed: 0

consumer waiting

Put: 2

Intimate consumer

producer waiting

creat: 2

Intimate Producer

consumed: 2

Put: 2

Intimate consumer

producer waiting

creat: 2

Intimate consumer

producer waiting

Put 3:

Intimate Waiting

creat: 3

Intimate consumer

consumed: 3

Put: 4

Intimate consumer

consumed: 4

Put: 5

dead locking →

class A d

synchronized void foo(B b) {

String name = Thread.currentThread().getNome();
System.out.println(name + " entered A.foo()");

} // anonymous class

try {

Thread.sleep(1000);

}

catch (Exception e) {

System.out.println("A interrupted");

}

System.out.println("A trying to call B.last()");
e.last();

}

void last() {

System.out.println("Inside A.last()");

}

class B d

synchronized void bar(A a) {

String nome = Thread.currentThread().getNome();

System.out.println(nome + " entered B.bar()");

try {

Thread.sleep(1000);

} // anonymous class

catch (Exception e) {

System.out.println("B interrupted");

System.out.println("inside A lock");
a.exit();

Y

void lost() {

System.out.println("inside A lock");

Y

class Deadlock implements Runnable {

A a = new A();

B b = new B();

Deadlock() {

Thread currentThread() { returnName("Main Thread"); }

Thread t = new Thread(this, "Racing Thread");

t.start();

a.join();

System.out.println("bar in main thread");

Y

public void run() {

b.join();

System.out.println("bar in other thread");

Y

public static void main(String args) {

new Deadlock();

Y

Y

Output is

- Main thread entered A-loc
- Racing thread entered B-bar
- Main thread trying to call B.last()
- Inside A.last
- Back in main thread
- Racing thread trying to call B.last()
- Inside A.last
- ~~Back in other thread~~

~~Fix~~
13.02.24