## VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.



LAB REPORT on

# Object Oriented Java Programming (23CS3PCOOJ)

Submitted by

Kishore Chandra N (1BM23CS154)

in partial fulfillment for the award of the degree of BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING



#### BENGALURU-560019 Sep-2024 to Jan-2025

#### B.M.S. College of Engineering,

Bull Temple Road, Bangalore 560019
(Affiliated To Visvesvaraya Technological University, Belgaum)

Department of Computer Science and Engineering



#### **CERTIFICATE**

This is to certify that the Lab work entitled "Object Oriented Java Programming (23CS3PCOOJ)" carried out by **Kishore Chandra N** (**1BM23CS154**), who is bonafide student of **B.M.S.** College of Engineering. It is in partial fulfillment for the award of Bachelor of Engineering in Computer Science and Engineering of the Visvesvaraya Technological University, Belgaum. The Lab report has been approved as it satisfies the academic requirements in respect of an Object Oriented Java Programming (23CS3PCOOJ) work prescribed for the said degree.

Dr. Jyothi S Nayak
Professor & HOD
Department of CSE, BMSCE

### Index

Muca			
Sl. No.	Date	Experiment Title	Page No.
1	30-9-2024	Quadratic Equations	4
2	7-10-2024	SGPA Calculator	7
3	14-10-2024	Book Details	10
4	21-10-2024	Area of Figures	13
5	28-10-2024	Bank Account Details	16
6	11-11-2024	SEE and CIE Marks Displayer	23
7	28-11-2024	Exception Handling	28
8	28-11-2024	Threads	32
9	28-11-2024	Divisor App	35
10	28-11-2024	Inter Process Communication and Deadlock	40

#### Github Link: https://github.com/Kishore868/OOJ-Lab-Programs

#### Program 1

Implement Quadratic Equation

Algorithm:

Lab Program.

1. Develop Java program that prints all real solution

to QE ax²+bx+c=0. Read in a,b,c and we
formula. If discriminant b²-4ac=-ve, display
message stating no real solutions.

A. import static gava. lang. Math. 29t;
import java. util. Scanner;

public class QE

S int a,b,c;
double a1,72,d;

```
void input()
      Scanner &c = new Scanner (System.in);
     System.out println ("Enter value of a: ");
     a = sc. next Int();
    while ( a == 0)
            System.out.printly ("Enter a non-zero number
                            for a: ");
           a = sc.neatInt();
           d=b*b-4*a*c;
      3
void display()
     14 (d==0) S
       n1= -b/(2.0 * a);
     System out-println ("Roots are real and equal");
      System.out-puntln (" Root: "+91);
    de of (d>0) {
    al = (-b+ squt(d))/(20 *a);
   92= (-b- squt(d)) / (20 * a);
   System out-punkly ("Roots are real and distinct");
   System. out-println (" 31="+91+", 12="+92);
    else {
    n1=-b/(2.0 * a);
    92= squt (-d)/(2.0 * a);
```

```
System.out. puntln ("Roots are ?maginary");
    System.out. puntln ("91="+91+"++92+").
   System.out-puntly ("92="+91+"-"+92+"1")
   System outpoon 3
  public static void main (String [] augs) 5
      GE ge = New GE();
      qe. Input();
      ac . display ();
  7
   Output:
  Enter value of a: 1 ... Enter value of a: 2
  Enter value of b: -4 Enter value of b: 2
  Enter value of c: 4
                          Enter value of c! 2
 Roots are real and equal Roots are imaginary
  Root: 2.0
                           91=0.5 t 0.8660;
 Enter value of a: 1
 Enter realise of b:-9
 Enter value of c: 10
 Roots are real and different
91=7.701, 12=1.2984 D= 3990pm and 1 de etc
Code:
import static java.lang.Math.sqrt;
import java.util.Scanner;
class QE {
  int a, b, c;
  double r1, r2, d;
  void input() {
  Scanner sc = new Scanner(System.in);
  System.out.print("Enter value of a: ");
     a = sc.nextInt();
     while (a == 0) {
        System.out.println("Enter a non-zero number for a:");
        a = sc.nextInt();
     }
    System.out.print("Enter value of b: ");
     b = sc.nextInt();
     System.out.print("Enter value of c: ");
     c = sc.nextInt();
```

```
d = b * b - 4 * a * c;
  }
void display() {
     if (d == 0) {
       r1 = -b / (2.0 * a);
       System.out.println("Roots are real and equal");
       System.out.println("Root: " + r1);
System.out.println("1BM23CS154");
       System.out.println("Kishore Chandra N");
     \} else if (d > 0) {
       r1 = (-b + sqrt(d)) / (2.0 * a);
       r2 = (-b - sqrt(d)) / (2.0 * a);
       System.out.println("Roots are real and different");
       System.out.println("r1 = " + r1 + ", r2 = " + r2);
System.out.println("1BM23CS154");
       System.out.println("Kishore Chandra N");
     } else {
       r1 = -b / (2.0 * a);
       r2 = sqrt(-d) / (2.0 * a);
       System.out.println("Roots are imaginary");
       System.out.println("r1 = " + r1 + " + " + r2 + "i");
       System.out.println("r2 = " + r1 + " - " + r2 + "i");
          System.out.println("1BM23CS154");
       System.out.println("Kishore Chandra N");
     }
 public static void main(String[] args) {
     QE qe = new QE();
     qe.input();
     qe.display();
  }
}
Output:
D:\1BM23CS154>java QE.java
```

```
D:\lBM23CS154>java QE.java
Enter value of a: 1
Enter value of b: -9
Enter value of c: 10
Roots are real and different
r1 = 7.701562118716424, r2 = 1.2984378812835757
1BM23CS154
Kishore Chandra N
```

```
D:\1BM23CS154>java QE.java
Enter value of a: 2
Enter value of b: 2
Enter value of c: 2
Roots are imaginary
r1 = -0.5 + 0.8660254037844386i
r2 = -0.5 - 0.8660254037844386i
1BM23CS154
Kishore Chandra N
```

```
D:\1BM23CS154>java QE.java
Enter value of a: 1
Enter value of b: -4
Enter value of c: 4
Roots are real and equal
Root: 2.0
1BM23CS154
Kishore Chandra N
```

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.

```
2. WAP to wrate class student with members. usin
name, an away wedits and away marks. Include
methods to accept & display details and method
to calculate SLAPA of a student.
A. import Java util Scanner:
 public also Student (
    string name, wn;
    double SGPA;
 int[] maks = new int[4];
  int[] condity 2 new int[4];
   double [] gradopoints - new double [4];
  double total = 0, credit total = 0;
  Scanner sc = New Scanner (System 3n);
  void getstudent Details () §
     System.out. println ("Enter name: ");
      name = sc. neathne();
     System. out . println ("Enter USN: ");
     wsn = st. neathine ();
  9
 void getMarks () S
     for ( Int 5=0; 124; 1++)
    System. out. pinter ("Enter" + (j+1) + Subject mak.)
         marks[j] = sc-next_Int();
      System out printly ("Enter wells for subjects"
                                 +(5+0+ ":");
```

```
(lodits[i] = sc. nextInt();
   gradeparints[i] = (mails[i]/10)t1;
   if (geadepoints [i]>10) {
       gladepoints [j] = 10!
 3
void compute SGPA() 5
 for (int 3 = 0; 124; 3++) $
     total + = gradepoints [i] + add [i];
     credit total + = credits []]
void display () {
    System out-puntly ("Name: "trame).
    System. out-printly ("USN: "+ USD).
    System.out puntln ("SUPA: "+SUPA);
 4
        static void main (string [] ourgs) {
         Scanner SC = new Scanner (System. in);
         System. But-println ("Enter number of students")
         Pot nos = sc. neat Int();
        Student [] students = new Student [now];
```

```
Box lint 1=0; 12 nos; 1++) {
       Students [i] = new Student();
       students [i]. get Student Details ();
       Students [i]. getMarks ();
      Students [1]. Compute SGPA();
       Students [8]. display();
    3
   System.
  Output: I Enter number of students!
 Enter name:
                            Name: KCN
 KCNI
                             USN: IBM 23C5001
 Enter USN:
                             SGPA: 9-24
 1BM23es001
Enter I subject marks:
  80
 Enter credits for subject 1:
  3
 Enter 2 subject marks:
 Exter wealth for suffect 2:
  2
 Enter 3 subject marks:
  90
Enter cuedits for subject 3:
Enter 4 Subject marks: 100 parts
 90
Enter credits for subject 4:
Code:
import java.util.Scanner;
public class Student {
  String name, usn;
  double SGPA;
  int[] marks = new int[4];
  int[] credits = new int[4];
  double[] gradepoints = new double[4];
  double total = 0, credittotal = 0;
  Scanner sc = new Scanner(System.in);
  void getStudentDetails() {
     System.out.println("Enter name:");
     name = sc.nextLine();
     System.out.println("Enter USN:");
     usn = sc.nextLine();
  void getMarks() {
     for (int j = 0; j < 4; j++) {
```

```
System.out.println("Enter" + (j + 1) + " subject marks:");
     marks[j] = sc.nextInt();
     System.out.println("Enter credits for subject " + (i + 1) + ":");
     credits[j] = sc.nextInt();
     gradepoints[j] = (\text{marks}[j] / 10.0) + 1;
     if (gradepoints[j] > 10) {
       gradepoints[i] = 10;
  }
  sc.nextLine();
void computeSGPA() {
  total = 0;
  credittotal = 0;
  for (int j = 0; j < 4; j++) {
     total += gradepoints[j] * credits[j];
     credittotal += credits[i];
  SGPA = total / credittotal;
void display() {
  System.out.println("Name: " + name);
  System.out.println("USN: " + usn);
  System.out.println("SGPA: " + SGPA);
}
public static void main(String[] args) {
  Scanner sc = new Scanner(System.in);
  System.out.println("Enter number of students:");
  int numberOfStudents = sc.nextInt();
  sc.nextLine();
  Student[] students = new Student[numberOfStudents];
  for (int i = 0; i < numberOfStudents; i++) {
     students[i] = new Student();
     students[i].getStudentDetails();
     students[i].getMarks();
     students[i].computeSGPA();
     students[i].display();
  }
              System.out.println("1BM23CS154 Kishore Chandra");
```

```
Enter name:
XYZ
Enter USN:
456
Enter 1 subject marks:
Enter credits for subject 1:
Enter 2 subject marks:
Enter credits for subject 2:
Enter 3 subject marks:
Enter credits for subject 3:
Enter 4 subject marks:
Enter credits for subject 4:
Name: XYZ
USN: 456
SGPA: 8.85
1BM23CS154 Kishore Chandra
```

```
\1BM23CS154>javac Student.java
):\1BM23CS154>java Student.java
nter number of students:
nter name:
CN
nter USN:
nter 1 subject marks:
Enter credits for subject 1:
Enter 2 subject marks:
nter credits for subject 2:
Enter 3 subject marks:
Enter credits for subject 3:
Enter 4 subject marks:
nter credits for subject 4:
lame: KCN
JSN: 1223
```

#### **Program 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.

```
3. Create class Book which contains for member,
 name, author, piece, num pages. Include constructor
 to set values for members. Include mathods
 to set and get the obtails of objects. Include
 a tosting () method that and display the
complete details of the book. Develop Torra
 pussion to create a book objects.
 A. import java. util. Scamer;
   class Books
          String name;
           String outher;
      nt price, numpages;
 Books (Sting name, String author, Pat pine, int numPages)
        this name = name
         this autro = author;
        this price = price;
        this numPages = NumPages;
 public String tosting ()
       String name, author, price, numbages:
      name = "Book name : "+ this name + "10";
     author = " Author name : " + this author + "10"
    pince . " Bice: " + this pine + " In".
    NumPager = Number of pages: "+ thi numPager = 41 m.
```

```
setur name + author + piece + numpage;
public static void main (swing args [7]
      Scanner sc=new Scanner (System.in);
      int n;
     String name;
    String outhor;
     ent puce;
   int numpages;
   System. out puntle ("Enter no. of books");
   n = sc.neatInt();
   Books b[];
   b = new Books [n];
   System out printly ("Enter book details");
   for (int 1=0; 1/1;1++)
        System out println ("Enter book "+(i+1)+" name");
        name = sc.next();
       System-out-println ("Enter book "+ (i+1)+ "outher")
     author = sc.next();
      System-out-println ("Enter book" + (1+1) + "purce")
       puce = sc-neatInt();
      System.out-println (" Enter book "+ (19+1)+ " pages ");
      numPages = SC-neatInt();
     b[1] = new Books ( name, author, pace, num Pages);
     for (int i=0; izn; i+t)
```

```
b[1]. toStiling ();
        System.out.println ( XXX b[i]);
   3
   Output :
   Enter no. of Books
  Enter book details
  Enter book I name
  Bharath
  Enter book | outhor
  Bhasken
  Enter book 1 piece
 35
  Enter Sook 1
  Book name: Bharath
 Author
 Number of pages: 72
Code:
import java.util.Scanner;
class Books
         String name;
         String author;
         int price,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()

int n; String name;

String name, author, price, numPages; name = "Book name: " + this.name + "\n"; author = "Author name: " + this.author + "\n";

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

public static void main(String args[])

return name + author + price + numPages;

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

Scanner sc=new Scanner(System.in);

```
String author;
       int price;
       int numPages;
       System.out.println("Enter no. of books");
       n=sc.nextInt();
       Books b[];
       b=new Books[n];
       System.out.println("Enter book details");
       for(int i=0;i< n;i++)
       System.out.println("Enter book "+(i+1)+" name");
       name=sc.next();
       System.out.println("Enter book "+(i+1)+" author");
       author=sc.next();
       System.out.println("Enter book "+(i+1)+" price");
       price=sc.nextInt();
       System.out.println("Enter book "+(i+1)+" pages");
       numPages=sc.nextInt();
       b[i]=new Books(name,author,price,numPages);
}
       for(int i=0;i<n;i++)
       b[i].toString();
       System.out.println(b[i]);
}
       System.out.println("Kishore Chandra N");
       System.out.println("1BM23CS154");
Output:
D:\1BM23CS154>javac Books.java
D:\1BM23CS154>java Books
Enter no. of books
Enter book details
Enter book 1 name
Bharath
Enter book 1 author
Bhasker
Enter book 1 price
35
Enter book 1 pages
72
Book name: Bharath
Author name: Bhasker
Price: 35
Number of pages: 72
Kishore Chandra N
1BM23CS154
```

Develop a Java program to create an abstract class named Shape that contains two integers and an empty methodnamed printArea(). Provide three classes named Rectangle, Triangle and Circle such that each one of theclasses extends the class Shape. Each one of the classes contain only the method printArea() that prints the area of the given shape.

Algorithm:

```
4. Develop plagram to whate abstract clay
                                           named Shape
that contains 2 Pritegess and empty method named
puntaval). Provide 3 classes named Rectangle, Trianglo
 and Circle such that each one of classes extends chos
 Shape. Each one of clases contain only method
punthereal) that prints area of given shape.
A. import Java. util. Scanner;
  abstract class shape I
         protected int dimension);
        protected int demension2;
     public do abstract void printArial);
   y from it puttle [ Alex of dull. In force)
        Rectangle extends Shape {
         public Rectangle (int langth, int width) {
               this dimension = length;
              this dimension 2 = wholth;
           public void pointAcea() {
             Int area = dimension 1 * dimension 2;
             System out println ("Area of Rectangle: "+area);
  class Triangle extends Shape &
           public Triangle (int base, int height) S
                 this dimension = base;
                this dimension 2 = height
```

```
public void printAlea () S
       double area = 0.5 x dimension 1 x dimension 2;
       System. out-println ("Area of Triangle: "+ area);
 3
 class
           circle extends shape §
         public Circle (int radius) S
                this . dimension 1 = radius;
        public void puntacea () S
            double alea = Math. PI + dimension 1 * dimension 1:
            System out printer ( Alea of circle: 10 tarea):
      3
3
public dass shape Test S
      public static vold main (String [] args)
               Scanner Sc = per Scanner (System.in);
            System out printly (" Enter length & breadth of
                               rectangle ").
              int dength = sc.nead Int();
              int breadle 2 Sc. neart Int ();
            Shape rectangle = new Rectangle (lenoth, breath)
             rectangle . printAlea ();
```

```
System. out println ("Enter base & height of trongle")
              int base = sc. nextInt();
              Int Leight = sc-nestInt();
             Shape triangle = new Triangle (base, height);
             tuangle pintAlea()
            System out puntln ("Enter radius of circle");
            int radius = sc. neutInt();
           Shape circle = new circle (radius);
           chicle print Area ();
     3 man words are had small with which of the others of the same to 
   Output:

Enter length & breadth of rectangle

3 2

Area of Rectangle: 6

Enter base & bright of triangle
     24
    Area of Triangle: 4.0
   Enter radius of circle:
     Alex of citle: 50.265
    dizzilio
                            - Transperie Porce - form
                                              The last transit
                                                                         Suffer Trader
Code:
import java.util.Scanner;
abstract class Shape {
       protected int dimension1;
       protected int dimension2;
       public abstract void printArea();
class Rectangle extends Shape {
       public Rectangle(int length, int width) {
               this.dimension1 = length;
               this.dimension2 = width;
        }
             public void printArea() {
              int area = dimension1 * dimension2;
               System.out.println("Area of Rectangle: " + area);
class Triangle extends Shape {
       public Triangle(int base, int height) {
               this.dimension1 = base;
               this.dimension2 = height;
        }
```

```
public void printArea() {
    double area = 0.5 * dimension1 * dimension2;
    System.out.println("Area of Triangle: " + area);
  }
class Circle extends Shape {
  public Circle(int radius) {
    this.dimension1 = radius;
  public void printArea() {
    double area = Math.PI * dimension1 * dimension1;
    System.out.println("Area of Circle: \n" + area);
       System.out.println("1BM23CS154 Kishore Chandra");
  }
public class ShapeTest {
  public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    System.out.print("Enter length of rectangle: ");
    int length = scanner.nextInt();
    System.out.print("Enter width of rectangle: ");
    int width = scanner.nextInt():
    Shape rectangle = new Rectangle(length, width);
    rectangle.printArea();
    System.out.print("Enter base of triangle: ");
    int base = scanner.nextInt();
    System.out.print("Enter height of triangle: ");
    int height = scanner.nextInt();
    Shape triangle = new Triangle(base, height);
    triangle.printArea();
    System.out.print("Enter radius of circle: ");
    int radius = scanner.nextInt();
    Shape circle = new Circle(radius);
    circle.printArea();
    scanner.close();
  }
Output:
D:\1BM23CS154>java ShapeTest
Enter length of rectangle: 3
Enter width of rectangle: 2
Area of Rectangle: 6
Enter base of triangle: 2
Enter height of triangle: 4
Area of Triangle: 4.0
Enter radius of circle: 4
Area of Circle:
50.26548245743669
1BM23CS154 Kishore Chandra
```

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. Create a class Account that stores customer name, account number and type of account. From this derive the classes Cur-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:

- a) Accept deposit from customer and update the balance.
- b) Display the balance.
- c) Compute and deposit interest
- d) Permit withdrawal and update the balance Check for the minimum balance, impose penalty if necessary and update the balance.

```
28/10/2024
5. WAP
 two
called
        savings account
                              compound interest and
Savings
                    provides
                          but no cheque book facility.
               faciliti es
 withdrawal
                           cheque book faillity but
        account provides
Current
       interest. Current
                        balance and of
 maintain
             กาใก้เกนก
                this
       lelow
- Create
          number
account
Cw-acct
 specifice
a · Accept
            balance
b. Display
              and deposit interest
c. Compute
            withdrawal and
d. Permit
            for minimum
          gara util. Scanner;
A. Proport
                   Account
            class
                            customer Name
          protected - String
2
                     Stung
                    double
         profeded
                        account Number;
```

```
Account (String name, int acc Number, String acctype)
          customer Name = name;
         account Number = acc Number
         account Type =
                         accType :
        balance = 0;
  public
                   deposit (double amount)
           balance + = amount;
5
          System.out · puntln ("Deposited: "+ amount + "Updated
 4
  public
         void display Balance ()
             System. out partin ("Account Balance: "+balance).
  3
                                 Operation is specific
    3
         Sav Account extends
class
5
        SavAccount (Stiling name, int accNumber
```

```
public
          Void compute Interes (()
         double interest = balance * interest Rate;
         balance + = interest;
        System out println ("Interest added: "+interest + "
                          " Updated balonce: "+ balance).
  3
public void withdraw (double amount)
                                                              3
          If (balance > = amount)
               balance - = amount;
            System. out. printly ("Withdrawn: "tamount+
                            " Updated Balonce " + balance).
       else
      [ System.out. println ("Insufficient balance );
                                                                   elso
   3
                                                                 . 3
class CurAccount eatends Account
         double MinBalanc = 500.0;
                                                            3
        double servicellarge = 50.0;
                                                            public
   EurAccount (String name, int aunumber)
               super ( name, acc Number, " (went ):
      9
```

```
System. out. printly ("Enter occount number");
    int accommber : sc. neartint ();
  Sav Account savings Account = New Sov Account frome,
System.out. println (" Enter unstomer name: ");
 String namel = sc. not();
System out println ("Enter account number");
 int accountingbell = sc. neat [ntl);
Cur Account curentAccount = New CurAccount (name 1, accounter to )
while (true)
   System out - printle (" In --- - MENU --- ");
    System set pintly ("1. Deposit \n2. Withdraw \n3. Compute
                    Interest In 4. Display In S. Exit )
   System. out. print ("Enter your chorce: ");
  int charce = sc. next_Int();
 System out . print ("Enter type of account: ").
 String acctype = sc nest ()
4 ( actype equals ( "saving ")
        switch (choice)
           Case 1: Systemout . pint ("Enter deposit
```

```
public vold check MinBalance()
         if (balance / minBalance)
              balance - = service charge;
             System and printly ("Balance below minimum
                         * Service charge "+ servicellarge +
                           " Updated balance : "+balane);
public void withdraw (double amount)
       If (balance > = amount)
            balance - = a mount;
          System and puntly ("Withdrawn: "+amount+" Updated before "
                            t balance):
          check MinBalance ()
            System.out. println ("Insufficient Balance ");
       class
              Brink
        public static void main (String[] args)
             Scanner Siz rew Scanner (System.in);
           System out printly ("Enter customer name")
           String name = sc. next();
 double depositAmount 2 st. neatDouble ():
Savings Account . deposit (deposit Amount) .
bleck;
```

```
case 2: System.out.peint ("Enter withdrawal amount").
            double withdrawalamount 2 Sc. newthauble ().
          Sangs Account . withdraw (with drawal Amount);
          breck;
  case 3: savings Account compute interes (();
        book;
 case 4: System out puntly ( - Customer none + 2000 yellicans).
          System out purtlal "Account number " +100 years and Oceanthan
        System and pinth ("Type of Acant: "+300 yellown account Type)
       saving Account display ochoniel,
       black;
case s: System east (0);
I else if (actype equals ("unent"))
    & sweet (whoice)
      { case 1: System, out. point ("Enter deposit amount: ")'
                double departant sc. next Double();
               Coulent Account. Lipost ( deportant);
              brok ;
```

```
Case 2: System.out. part ("Enter withdrawar arount y.
            double withdrawal Amt = sc. next Double ();
           currentAccount . cottodiano (withdrawar/Amt);
           bleak;
 case 3. System out puntled "launt accounts do not
                              cour interest ");
             beak;
case 4: System and printle ("Name: "+ ament Account. ( with reviber)
         System out purilly ( " Account Number: "+ would Account account Number
       System out puntly ("Type of Alexant" + envent Acoust account Type)
      content Account display Balance ();
      bleak;
           System. east (0);
case 5:
            beak;
           System out - print (" Invalid account type ").
```

```
I. Pepsyit
2. Withchaw
3. Compile Interest
4. Display
5. Esit
Enter your choice: 3
Enter type of occurry: Saving
Interest added. 32.0. Updated bakene = 832.0
```

```
Output !
 Enter automer 1 Name ;
 Thishole
 Enter account number;
 1234
Enter custamer name:
 Kislan
 Enter account number:
 890
 ---- MENU --- .
1. Deposit
2. Withdraw
3. Compute Interest
4. Display
5, Exit
Enter your doice
Enter type of account : saving
Enter deposit amount: 1000
Deposited: 1000.0 Updated Idane: 1000.0.
1. Deposit.
2. Withdraw
3- compute Interest
4. Display
J. Eit
Enter you choice
Files type of account : saving
Enter withdrawal amount: 200
Withdrawn: 200.0 Updated bollance: 800.0
```

#### Code:

```
import java.util.Scanner;
class Account {
  String customerName;
  int accountNumber;
  String accountType;
  double balance;
  Account(String name, int accNumber, String accType) {
    customerName = name;
    accountNumber = accNumber;
    accountType = accType;
    balance = 0;
  }
  public void deposit(double amount) {
    balance += amount;
    System.out.println("Deposited: " + amount + ". Updated balance: " + balance);
  public void displayBalance() {
    System.out.println("Account Balance: " + balance);
  }
  public void withdraw(double amount) {
    System.out.println("This operation is specific to account type.");
  }
class SavAccount extends Account {
  double interestRate = 0.04; // 4% annual interest rate
  SavAccount(String name, int accNumber) {
    super(name, accNumber, "Savings");
  public void computeInterest() {
    double interest = balance * interestRate;
    balance += interest;
    System.out.println("Interest added: " + interest + ". Updated balance: " + balance);
  public void withdraw(double amount) {
    if (balance >= amount) {
       balance -= amount;
       System.out.println("Withdrawn: " + amount + ". Updated balance: " + balance);
     } else {
       System.out.println("Insufficient balance.");
```

```
}
class CurAccount extends Account {
  double minBalance = 500.0;
  double serviceCharge = 50.0;
  CurAccount(String name, int accNumber) {
    super(name, accNumber, "Current");
  }
  public void checkMinBalance() {
    if (balance < minBalance) {
       balance -= serviceCharge;
       System.out.println("Balance below minimum. Service charge imposed: " + serviceCharge + ". Updated
balance: " + balance);
  }
  public void withdraw(double amount) {
    if (balance >= amount) {
       balance -= amount;
       System.out.println("Withdrawn: " + amount + ". Updated balance: " + balance);
       checkMinBalance();
       System.out.println("Insufficient balance.");
    }
  }
public class Bank {
  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter customer name:");
    String name=sc.next();
    System.out.println("Enter account number:");
    int accountnumber=sc.nextInt();
    SavAccount savingsAccount = new SavAccount(name, accountnumber);
System.out.println("Enter customer name:");
    String name1=sc.next();
    System.out.println("Enter account number:");
    int accountnumber1=sc.nextInt();
    CurAccount currentAccount = new CurAccount(name1, accountnumber1);
    while (true) {
       System.out.println("\n----MENU-----");
       System.out.println("1. Deposit\n2. Withdraw\n3. Compute Interest for Savings Account\n4. Display
Account Details\n5. Exit");
       System.out.print("Enter your choice: ");
       int choice = sc.nextInt();
       System.out.print("Enter the type of account (saving/current): ");
       String accType = sc.next();
```

```
if (accType.equals("saving")) {
  switch (choice) {
    case 1:
       System.out.print("Enter the deposit amount: ");
       double depositAmount = sc.nextDouble();
       savingsAccount.deposit(depositAmount);
       break:
    case 2:
       System.out.print("Enter the withdrawal amount: ");
       double withdrawalAmount = sc.nextDouble();
       savingsAccount.withdraw(withdrawalAmount);
       break:
    case 3:
       savingsAccount.computeInterest();
       break:
    case 4:
       System.out.println("Customer name: " + savingsAccount.customerName);
       System.out.println("Account number: " + savingsAccount.accountNumber);
       System.out.println("Type of Account: " + savingsAccount.accountType);
       savingsAccount.displayBalance();
       break:
    case 5:
       System.exit(0);
       break;
    default:
       System.out.println("Invalid choice.");
} else if (accType.equals("current")) {
  switch (choice) {
    case 1:
       System.out.print("Enter the deposit amount: ");
       double depositAmount = sc.nextDouble();
       currentAccount.deposit(depositAmount);
       break;
    case 2:
       System.out.print("Enter the withdrawal amount: ");
       double withdrawalAmount = sc.nextDouble();
       currentAccount.withdraw(withdrawalAmount);
       break:
    case 3:
       System.out.println("Current accounts do not earn interest.");
       break:
    case 4:
       System.out.println("Customer name: " + currentAccount.customerName);
       System.out.println("Account number: " + currentAccount.accountNumber);
       System.out.println("Type of Account: " + currentAccount.accountType);
       currentAccount.displayBalance();
       break;
    case 5:
       System.exit(0);
       break;
    default:
```

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

} else {
System.out.println("Invalid account type.");
}

}
}
```

```
D:\1BM23CS154>java Bank
                                                                                                      --MENU----
Enter customer name:
Kishore
                                                                                                     Deposit
                                                                                                     Withdraw
Enter account number:
1234

    Compute Interest for Savings Account

                                                                                                4. Display Account Details
5. Exit
Enter customer name:
Kishen
Enter account number:
                                                                                                Enter your choice: 1
Enter the type of account (saving/current): current
Enter the deposit amount: 2000
Deposited: 2000.0. Updated balance: 2000.0
890
       -MENU--
1. Deposit
2. Withdraw
    Compute Interest for Savings Account
Display Account Details
                                                                                                 L. Deposit
5. EXIT
Enter your choice: 1
Enter the type of account (saving/current): saving
Enter the deposit amount: 1000
Deposited: 1000.0. Updated balance: 1000.0

    Withdraw
    Compute Interest for Savings Account

                                                                                                4. Display Account Details
                                                                                                5. Exit
                                                                                                enter your choice: 2
Enter the type of account (saving/current): current
Enter the withdrawal amount: 200
       -MENU-
1. Deposit
2. Withdraw
3. Compute Interest for Savings Account
4. Display Account Details
5. Exit
                                                                                                 Vithdrawn: 200.0. Updated balance: 1800.0
Enter your choice: 2
Enter the type of account (saving/current): saving
Enter the withdrawal amount: 200
Withdrawn: 200.0. Updated balance: 800.0
                                                                                                      --MENU--
                                                                                                     Deposit
                                                                                                     Withdraw
                                                                                                     Compute Interest for Savings Account
                                                                                                     Display Account Details
       -MENU-
1. Deposit
2. Withdraw
                                                                                                5. Exit
                                                                                                nter your choice: 3
Enter the type of account (saving/current): current
Current accounts do not earn interest.
    Compute Interest for Savings Account
Display Account Details
Exit
Enter your choice: 3
Enter the type of account (saving/current): saving
Interest added: 32.0. Updated balance: 832.0
                                                                                                      --MENU--
                                                                                                 l. Deposit
                                                                                                     Withdraw
       -MENU-
-----MENU-----

1. Deposit

2. Withdraw

3. Compute Interest for Savings Account

4. Display Account Details

5. Fyit
                                                                                                     Compute Interest for Savings Account
                                                                                                 1. Display Account Details
                                                                                                 5. Exit
                                                                                                Enter your choice: 4
Enter the type of account (saving/current): current
Customer name: Kishen
5. Exit
Enter your choice: 4
Enter the type of account (saving/current): saving
Customer name: Kishore
Account number: 1234
Type of Account: Savings
Account Balance: 832.0
                                                                                                Account number: 890
Type of Account: Current
Account Balance: 1800.0
```

Create a package CIE which has two classes- Student and Internals. The class Personal has members like usn, name, sem. The class internals 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.

```
6. Craste a package CIE which has a clause,
                                                               public void displaystudent ()
  Students & Internal . The class student has markers
                                                                     System.out. println ("USN: "+USN);
  like USN, name, sem class intende decinal from
                                                                     System out prently ("Nome: "+ name);
 Student has an away that stores steenal marks
             courses of current senester of student
        another package SEE with has to down
           which is a derived class of student
               away that stores SEE marks
 External
         in 5 some of current sem of student
 Class
                                                           11 Internals - java
          2 packages in a file that declares
                        students in all live courses
Import
                                                                   Java. util. Scanner;
                                                          public class Internals extends student
         CIE;
                                                                   protected int marks [] = new int[5];
       java. util. Scanner;
impod
                                                               public void input CIE marks ()
public
                                                                      Scanner sc = New Scanner (System. in)
       protected
                 String un;
                 string name;
                                                                     System. ait. println (" Enter internal marks
      protected
                int sem;
                                                                                      for 5 courses ")
      protected
                                                                    for (int 1=0; 125; 1++)
           state void input Student De talls ()
           Scarrer sc = new Scarner (System. b);
                                                                            System. out. print (" Enter marks for
          Systemout. printly ("Enter USN")
                                                                                           course "+(i+1)+": ");
          usn = & neathine ();
                                                                            marks [i] = sc. neat Int ();
          System. out - putatin ( " Enter Name : "):
         name = sc. nextline ();
        System out println ("Enter semester: ");
        sem=sc.neutinf();
```

```
public void displayFinalMooks ()
 11 Externals · java
                                                                                                                              display Student Details ();
package SEE;
Import CIE. Internals;
                                                                                                                             System. out. pointln ("Final Marks: ");
 public class Externals extends Internals
                                                                                                                           for (int 1=0; 925; 1+1)
              protected int marks[] = new int[5];
                                                                                                                                     System. out-pointh ("(oune "+(1+1)+": "+ find Houte
             protected at familyarks [] = new int[5];
             public Externals ()
                                                                                                               3
                       marks = new int[5];
                                                                                                               11 Moin.java
                      final Marks = new int [5];
                                                                                                              import SEE. Externals;
                                                                                                              import java-util Scanner;
          public void inputSEFMacks()
                    Scanner siz new Scanner (System. 9n);
                                                                                                              cleus Moun
                   System out printly ("Enter SEE marks for
                                                                                                                             public static vord marn (sturng augs[])
                            5 courses 9;
                 for (Pot 1=0; 125; 1++)
                                                                                                                         Scanner Sc=new Scanner (System. 9n):
                          System. ait. puint ("Enter SEE marks for
                                                                                                                                 · System.out.punt ("Enter no. of students");
                                              course "+(1+1)+": ");
                                                                                                                                      Put n = sc. neat Int()
                         marks [2] = sc.nextInt();
                                                                                                                                     Externals[] s = new (Externals[n];
                                   I then I be the design to
                                                                                                                           for (Bat i= 0; ix n; i++)
  public void calculate Final Marks ()
  & for (int 1=0; 915; 9++)
                                                                                                                                        $ s[i] = new Externals();
                    final Marks[i] = marks[i] t(this. marks[i])
                                                                                                                                      System. out . printeln (" Enter details of
                                                                                                                                                                           student "+(i+1)):
   3
                                                                                                                                    Stude S [9]. Input Student Details U;
                                                                                                                                                s[i]. Input (IEMarks1);
                   S[1] . InputSEEMarks )
                                                                                                                                       SEE marks for 5 courses.
                                                                                                                        Enter
                   s[i]. calculate Final Mouts();
                                                                                                                         70
                                                                                                                         70
                                                                                                                         80
       System. out-pently ("In Displaying Grad marks ").
                                                                                                                        90
     Box (int i=0; PKn; P++)
                                                                                                                       90
                                                                                                                    Displaying find marks for all students
                        system. out. printly ("Instrudent "+(i+1)+":").
                         S[i]. display Final Marks();
                                                                                                                      Student 1:
                                                                                                                    USN : IBMZ3CSISY
       14
                                                                                                                     Name! Kishole Chandra N
 3
                                                                                                                      Semester: 3
  Odput:
                                                                                                                    Final Marks: (Internal + Enternal)
 Enter no. of students ! 1
                                                                                                                      Course 1: 140
Enter details for student 1
 Enter USM: 18M2365154
                                                                                                                      Louise 2: 140
 Enter Name: Kishore Chancha
                                                                                                                     Louise 3: 160
 Enter semester: 3
                                                                                                                     Course 4: 180
Enter Internal Marks for 5 courses:
Enter marks for course 1: 30
Enter mades for course 2:38
                                                                                                                                               The state of the s
Enter marks but course 3:39
Enter marks Box course 4:37
Enter marks but course 5: 39
```

```
Code:
package SEE;
import CIE.Internals;
import java.util.Scanner;
public class Externals extends Internals {
  protected int marks[] = new int[5];
  protected int finalMarks[] = new int[5];
  public Externals() {
     marks = new int[5];
     finalMarks = new int[5];
  }
  public void inputSEEmarks() {
     Scanner s = new Scanner(System.in);
     System.out.println("Enter SEE Marks for 5 courses:");
     for (int i = 0; i < 5; i++) {
       System.out.print("Enter SEE marks for course " +(i + 1) + ": ");
       marks[i] = s.nextInt();
     }
  }
  public void calculateFinalMarks() {
     for (int i = 0; i < 5; i++) {
       finalMarks[i] = marks[i] + this.marks[i];
     }
  }
  public void displayFinalMarks() {
     displayStudentDetails();
     System.out.println("Final Marks (Internal + External):");
     for (int i = 0; i < 5; i++) {
       System.out.println("Course" + (i + 1) + ":" + finalMarks[i]);
  }
package CIE;
import java.util.Scanner;
public class Internals extends Student {
  protected int marks[] = new int[5];
  public void inputCIEmarks() {
     Scanner s = new Scanner(System.in);
     System.out.println("Enter Internal Marks for 5 courses:");
     for (int i = 0; i < 5; i++) {
```

System.out.print("Enter marks for course " + (i + 1) + ": ");

```
marks[i] = s.nextInt();
  }
package CIE;
import java.util.Scanner;
public class Student {
  protected String usn;
  protected String name;
  protected int sem;
  public void inputStudentDetails() {
     Scanner s = new Scanner(System.in);
    System.out.print("Enter USN: ");
    usn = s.nextLine();
     System.out.print("Enter Name: ");
     name = s.nextLine();
    System.out.print("Enter Semester: ");
    sem = s.nextInt();
  }
  public void displayStudentDetails() {
     System.out.println("USN: " + usn);
    System.out.println("Name: " + name);
     System.out.println("Semester: " + sem);
}
import SEE.Externals;
import java.util.Scanner;
class Main {
  public static void main(String[] args) {
     Scanner scanner = new Scanner(System.in);
     System.out.print("Enter the number of students: ");
     int n = scanner.nextInt();
     Externals[] students = new Externals[n];
     for (int i = 0; i < n; i++) {
       students[i] = new Externals();
       System.out.println("\nEnter details for student " + (i + 1));
       students[i].inputStudentDetails();
       students[i].inputCIEmarks();
       students[i].inputSEEmarks();
```

```
students[i].calculateFinalMarks();
     System.out.println("\nDisplaying final marks for all students:");
     for (int i = 0; i < n; i++) {
       System.out.println("\nStudent " + (i + 1) + ":");
       students[i].displayFinalMarks();
     }
  }
}
```

```
Enter the number of students: 2
Enter details for student 1
Enter USN: 1BM23CS154
Enter Name: Kishore Chandra N
Enter Semester: 3
Enter Internal Marks for 5 courses:
Enter marks for course 1: 30
Enter marks for course 2: 38
Enter marks for course 3: 39
Enter marks for course 4: 37
Enter marks for course 5: 39
Enter SEE Marks for 5 courses:
Enter SEE marks for course 1: 70
Enter SEE marks for course 2: 70
Enter SEE marks for course 3: 80
Enter SEE marks for course 4: 90
Enter SEE marks for course 5: 98
Enter details for student 2
Enter USN: 1BM23CS155
Enter Name: Kishen
Enter Semester: 3
Enter Internal Marks for 5 courses:
Enter marks for course 1: 40
Enter marks for course 2: 34
Enter marks for course 3: 36
Enter marks for course 4: 37
Enter marks for course 5: 38
Enter SEE Marks for 5 courses:
Enter SEE marks for course 1: 90
Enter SEE marks for course 2: 98
Enter SEE marks for course 3: 89
Enter SEE marks for course 4: 97
Enter SEE marks for course 5: 80
Displaying final marks for all students:
Student 1:
USN: 1BM23CS154
Name: Kishore Chandra N
Semester: 3
Final Marks (Internal + External):
Course 1: 140
Course 2: 140
Course 3: 160
Course 4: 180
Course 5: 196
Student 2:
USN: 1BM23CS155
Name: Kishen
Semester: 3
Final Marks (Internal + External):
Course 1: 180
Course 2: 196
Course 3: 178
Course 4: 194
Course 5: 160
```

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 uses both father and son's age and throws an exception if son's age is >=father's age.

#### Algorithm:

7. WAP that demonstrates in Interstance tree Create base class Son extends Father class called and derived class called "Son" which son Age; Int the base class. In Father class, implement public Son (int father Age, int son Age) through when gage a constructor which takes age and through Father Super (father Age); Wrong Age () when If (son Age > = father Age) that uses both class, implement constructor new wrong age ("Son's ex cannot be & son's age and throws an exception if greater than Father's sen's age is >= father's A. import java. util. Scanner; this son Age = son Age; System.out.puntln["son's age: "+sonAge); law Wrong Age (String message) 3 super (mossage) 2 3 class Main public staffic vold main (stunges agos) 3 public 4 { Scanner Sc=new Scanner (System=9n); class Father 3 Pnt System. out puntilly (" Enter Father's age Father (int age) throws public y (age >0) int father Agel = Sc. neat Int(); System.out-puntln (" Enter Son's age for 3 3 catch (Wrong Age e) s this age = age System out puntle ("Father's age: System. out. puntln ("Exception in Test case 1: 4 + e. get Mexage ()). الم ا

```
Systemout punth ("Enter father age for Test one 2:").
       int fatherage 2 = sc. next Int();
      Father father 2 = new Father (father Agree);
   y cotch (Wrong Age e) &
        Systemout. puntle ("Exception in Test case 1:"
                                  te.getMessage ());
       System. out. println ("Enter father's age box
  try &
                       Test case 3: 9;
      int father 19e3 = sc. neat Int ();
      System. out. println (" Enter son age ");
      int son Age 3 = Sc. neat Int ();
       Son son3 = new Son (pathinge3, sonAge3);
   I catch (Wrong Age e) {
         System out-puntly ( Exception: "+e-getMessage())
         finally [
     3 Outst: Sc. close ():
3 Enter Fother's age for Test Care 1:
          23
          Entel Son's age for Ted Case 2:
          Father's age: 23
         son's age: 12
Enter Father's age:
        Exception: Fotter a age cannot be regotive Enter Fatter & age: 30
Enter Som's age: 95
      Exception: son's age cannot be gleater than Father's age
Code:
import java.util.Scanner;
class WrongAge extends Exception {
   public WrongAge(String message) {
      super(message);
}
class Father {
  int age;
public Father(int age) throws WrongAge {
      if (age < 0) {
         throw new WrongAge("Father's age cannot be negative.");
```

```
}
    this.age = age;
    System.out.println("Father's age: " + age);
  }
class Son extends Father {
  int sonAge;
  public Son(int fatherAge, int sonAge) throws WrongAge {
    super(fatherAge);
    if (sonAge >= fatherAge) {
       throw new WrongAge("Son's age cannot be greater than or equal to Father's age.");
     }
    this.sonAge = sonAge;
    System.out.println("Son's age: " + sonAge);
  }
}
public class Main {
  public static void main(String[] args) {
     Scanner scanner = new Scanner(System.in);
try {
       System.out.println("Enter Father's age for Test Case 1:");
       int fatherAge1 = scanner.nextInt();
       System.out.println("Enter Son's age for Test Case 1:");
       int sonAge1 = scanner.nextInt();
       Son son1 = new Son(fatherAge1, sonAge1);
```

```
} catch (WrongAge e) {
       System.out.println("Exception in Test Case 1: " + e.getMessage());
     }
    try {
       System.out.println("\nEnter Father's age for Test Case 2:");
       int fatherAge2 = scanner.nextInt();
       Father father2 = new Father(fatherAge2);
     } catch (WrongAge e) {
       System.out.println("Exception in Test Case 2: " + e.getMessage());
     }
  try {
       System.out.println("\nEnter Father's age for Test Case 3:");
       int fatherAge3 = scanner.nextInt();
       System.out.println("Enter Son's age for Test Case 3:");
       int sonAge3 = scanner.nextInt();
       Son son2 = new Son(fatherAge3, sonAge3);
     } catch (WrongAge e) {
       System.out.println("Exception in Test Case 3: " + e.getMessage());
     } finally {
       scanner.close();
     }
  }
}
```

```
D:\lBM23CS154>java Main
Enter Father's age for Test Case 1:
23
Enter Son's age for Test Case 1:
12
Father's age: 23
Son's age: 12
Enter Father's age for Test Case 2:
-9
Exception in Test Case 2: Father's age cannot be negative.
Enter Father's age for Test Case 3:
30
Enter Son's age for Test Case 3:
45
Father's age: 30
Exception in Test Case 3: Son's age cannot be greater than or equal to Father's age.
```

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

```
class mead Rint
8. WAP which weater two threads, one thread
                                                            public static void main (String[] angs)
                                                       public
  displaying "BMS college of Engineering" one every
  ten seconds and another desplaying "CSE" DACE
                                                                 College Thread college Thread = new College Thread []!
                                                                CSETTHEOR CSETTHEORD ();
  every 2 seconds.
                                                               college Thread . start();
  A. class CollegeThiead calends Thread
                                                               cseThread.start();
         public void run ()
                                                                      A Albert Log De
            whele (true)
                                                        3
         ĺ
              try &
                                                      Output :
              System. out. plintln ("BMS College of Engineering"),
                                                      BMS college of Engineein 9
              Thread. sleep (10000);
                                                     CSE
         I cotd (Interrupted Exception e)
                                                     CSE
              System. out · puntln (e)
        Į
                                                     1.SE
                                                     CSE
        3
                                                     CSE
                                                     BMS college of Engineering
     3
                                                     CSE
                         cores (Warmangage e) &
   3
  3
        CSE Thread extends Thread
                                                     CSE
                                                      (SE
 class
                                                     CSE
        public void run()
                                                     Aus College of Engineering
              while (true)
                System. out. printer ("CSE");
                                                               Lateral last of and a million, and
                Thread sleep (2000)
            3 catch ( Interupted Exception c) 5
               System. out . puntln (e);
                                                                           with the man - take
          3 4
    4 3
```

```
Code:
```

```
class CollegeThread extends Thread {
  public void main{
    while (true) {
       try {
         System.out.println("BMS College of Engineering");
         Thread.sleep(10000);
       } catch (InterruptedException e) {
         System.out.println(e);
       }
     }
class CSEThread extends Thread {
 public void run() {
     while (true) {
      try {
         System.out.println("CSE");
         Thread.sleep(2000);
       } catch (InterruptedException e) {
         System.out.println(e);
```

```
public class ThreadPrint {
    public static void main(String[] args) {
        CollegeThread collegeThread = new CollegeThread();
        CSEThread cseThread = new CSEThread();
        collegeThread.start();
        cseThread.start();
    }
}
```

```
BMS College of Engineering
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
```

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 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box.

```
9. WAP to cleate user interface to perform
integer divisions. Division of Num! & Num?
is displayed in Result Gold, if Num & or
Num2 & not an integer, peoplar would theore
   Number Format Exception. If Num2 were zons
 program throws an Authoretic Exception.
A. impast jouax-swing. *;
   import java aut *;
   import . java. and event. *;
   class swing Demo {
  Swing Demo () {
   JErame Ifim = New JErame ("Divider App")
   jfum : set Size (295,150);
  I fim . set Defaut lesse Operation ( JErame . EXIT_ON_ CLOSE);
  Thatal j-lab = new Itabel ("Enter the divided and divident: ");
  I Tentfield afith = new ITendfield(8);
  JTextfield brith = new JTextfield(8);
  JButton Soutton = new JButton ("Calculate");
  Itabel er = new Itabel();
         allab = new JLabel ();
```

```
Itabel Stab = new Itabel ():
 Itabel andab = new Itabel ().
 Ifem. add (our);
jfim add (jlab);
ifum add (aits):
i fem . add ( bith);
ifin add (button);
fun add (alab);
fum add (blab):
fim-add (anslab)
                         ActionListener()
ActionListener | = new
      public void actionleyouned (ActionEvent evrl)
        System. out. puntln ("Action event from text field 4); 7,
    3;
   ajtf. add Action Listener (1)
  byth. add Action Literer (1).
  button-add Actionhisterer ( new Action listener 1)
         public void action Reformed (Action Event evt)
       { ty !
           int a = Integer . parse Int (a) + f . get Text()))
          int b = Integer. pause Int (bitf. getTeat());
          int and = a/b.
         alab. set Test ("InA = "ta):
         blab. set Tent ("n13 = " + b).
         andlab.setTeat ("In Anu = "tanu); y
```

```
catch (Number Format Exception e)

{ alab. set Teart ("");
       blab . set Text ("");
       anslab. setTeat ("");
      err. set Text ("B should be non reco! 9; y
   3);
   ifim . set Visible (true);
 Public state void main (Stung args [])
       Swing Utilities invokehater (new Runnable ()
     1
3
```

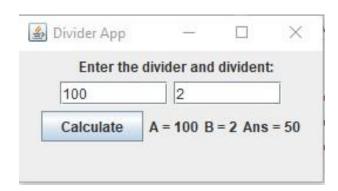
#### Code:

```
import javax.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());
// to terminate on close
jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
// text label
JLabel jlab = new JLabel("Enter the divider and divident:");
// add text field for both numbers
JTextField ajtf = new JTextField(8);
JTextField bjtf = new JTextField(8);
// calc button
JButton button = new JButton("Calculate");
// labels
JLabel err = new JLabel();
JLabel alab = new JLabel();
JLabel blab = new JLabel();
JLabel anslab = new JLabel();
// add in order :)
jfrm.add(err); // to display error bois
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(1);
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!"); }
}
});
// display frame
jfrm.setVisible(true);
}
public static void main(String args[]){ // create frame on event dispatching thread
SwingUtilities.invokeLater(new Runnable(){
public void run(){
new SwingDemo();
}
});
}
```



### Program 10

Demonstrate Inter process Communication and deadlock

# Algorithm:

```
10a. Inter Process Communication
A. class Q {
     int n;
    boolean value Set = false;
   synchronized int get ()
         whele (! valueSet)
     try &
     System.out.pointln("In Consumer waiting In 9;
    wait ();
   } cotch (Interrupted Exception e) {
   System out puntly ("Interrupted Exception caught");
  System.out.println ("Got: "+n)
  Value Set = false
  System. out. println ( "In Intimate Broduce In 4);
  noteby();
  return n;
  3
  synchronized void put (Int n)
 { while (value Set)
   System out punter (" Producer Waiting In ")
   wait();
    catch [ Interrupted Exception e) S
```

```
System.out.pintln ("Interrepted Exception caught");

3
this.n=n;

ValueSet = true;

System.out.pintln ("Put: "tn);

System.out.pintln ("Intimate Consumer In");

Notify();

3
class Producer implements Runnable of

R q;

Consumer (Q q)

{ this.q=p;

New Thread (this, "Consumer").start();

public void run()

{ nt i=0;

while (ix1s)

{ int x=q.get();

System.out.pintln ("Consumed: "4n);

8++;

3

3

4
```

```
Intimate
                                                                 Consume
      Intimate Produce
                                                      Got 6
       Put : 3
                                                     Intimate Rudua
     Intimate Consumer
                                                      cosumed: 6
                                                      Put: 7
     Produce woulting
                                                     Intimate Produces
     cosumed ! 2
                                                      & Rioduus waiting
     Got: 3
                                                      Got: 7
    Intimate Zuduces
                                                      Intimate Bloducy
    Put: 4
                                                      consumed: 7
    Cosumed: 3
                                                      Rut: 8
    Intimate Consumer
                                                      Intimate Consumos
                                                      Reduce waiting
   Leoduce waiting
                                                      Grot : 8
   Grot: 4
   Intimate Produces
    consumed : 4
   Put: 5
   Intimate Consumer
   Produce walting
   Got: 5
  Intimater Ploducy
  wasuned : 5
  Pu+: 6
Code:
class Q {
int n;
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("\nIntimate 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("\nIntimate Consumer\n");
notify();
```

```
} }
class Producer implements Runnable {
Qq;
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 {
Qq;
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 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:

```
D:\1BM23CS154>javac PCFixed.java
D:\1BM23C5154>java PCFixed
Press Control-C to stop.
Put: 0
Intimate Consumer
Producer waiting
Sot: 0
Intimate Producer
Put: 1
Intimate Consumer
Producer waiting
Got: 1
Intimate Producer
consumed:1
Put: 2
Intimate Consumer
Producer waiting
Got: 2
Intimate Producer
Put: 3
Intimate Consumer
Producer waiting
consumed:2
Got: 3
Intimate Producer
Put: 4
consumed:3
```

```
Intimate Consumer
roducer waiting
Got: 4
Intimate Producer
consumed:4
Put: 5
Intimate Consumer
Producer waiting
Got: 5
Intimate Producer
consumed:5
Intimate Consumer
Got: 6
Intimate Producer
consumed:6
Intimate Consumer
Producer waiting
Got: 7
Intimate Producer
consumed:7
Intimate Consumer
Producer waiting
```

# **Deadlock**

# Algorithm:

```
b. Deadlock.
                                                        System. out println ("B interupted "):
      class A S
     synchronized vold foo (B b) {
                                                       System.out. puntln (name + " trying to call 1. last () ").
    String name = Thread unentThread (). getName ();
                                                       a. last ();
   System out-pintln ( name + " entered A. 600 "):
                                                       3
                                                       void last()
  my ?
                                                            System. out. println ( "Inside A. last );
    Thread. sleep (1000).
  3 catch (Exception e) S
                                                      3
                                                      class Deadlack Implements Runnable
   System.out. puntln ("A Interrupted ")'
                                                          A a = new A();
  System-out-publish (name + " tying to call B. last();
                                                         B b = new B();
  b. last();
                                                        Deadlock () S
  3
                                                        Thread · wwent Thread () · set Name ("Mag Thread");
void last ()
                                                       Thead t = new Thread (this, "RailingThread");
      System.out. pontin ("Inside A. last");
4
                                                       t-start ();
3
                                                       a. foo(b);
class B S
                                                      System. out prently ("Back in main thready).
Synchronized void bar (A a) S
                                                      public void run ()
String name = Thread .curent Thread (). getName ();
                                                           b. bar (a);
System. out-partly (name + " entered B. bay ");
                                                       System. out-println (" Back in other thread ");
try {
  Thread - sleep (1000);
 y catch (Exception e) 5
```

```
b public static void main (Stiling args[])

{ new Deadlock ();
      3
     Octput:
     Mainthood entered A. foo
    Racing Thread entered B. bar
    MainThread trying to call B. last ()
   RacingThread trying to call A.last()
    Back in main thread
   Inside A last
   Back In other thread.
   de alialas sono senestro de manos remandos
Code:
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 {
synchronized 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();
void last() {
System.out.println("Inside A.last");
}
class Deadlock implements Runnable
A a = new A();
```

```
Deadlock() {
Thread.currentThread().setName("MainThread");
Thread t = new Thread(this, "RacingThread");
t.start();
a.foo(b); // get lock on a in this thread.
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();
}
D:\1BM23C5154>java Deadlock
MainThread entered A.foo
RacingThread entered B.bar
MainThread trying to call B.last()
Inside A.last
RacingThread trying to call A.last()
Back in main thread
Inside A.last
Back in other thread
```

B b = new B();