B.M.S COLLEGE OF ENGINEERING BENGALURU

Autonomous Institute, Affiliated to VTU

LAB REPORT

23CS3PCOOJ

Submitted in partial fulfilment of the requirements for Lab

Bachelor of Engineering

in

Computer Science and Engineering

Submitted by:

NIDHI A

1BM22CS177

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

College of Engineering,

Bull Temple Road, Basavanagudi, Bangalore, 560 019 2023-2024.

Index

Sl. No.	DATE	Experiment Title
A	**12/12/23-20/02/24**	**Scanned copy of Observation book**
1	12/12/23	Lab-01 Quadratic
2	19/12/23	Lab-02 Student SGPA calculations
3	26/12/23	Lab-03 Book program
4	02/01/24	Lab -04 Shape program
5	09/01/24	Lab-05 Back account (Abstract class)
6	16/01/24	Lab-06 Packages
7	23/01/24	Lab-07 Exceptions
8	30/01/24	Lab-08 Threads
9	06/01/24	Lab-09 Create user interface
10	20/01/24	Lab-10 IPC and Deadlock

Page No.
Date: / /

Features of Java.

I

1. Inspired by C/C++
2. Object Oriented
3. Platform independent
4. Multi - threaded
5. Dynamic
6. Robust
7. Secure
8. High - Performance
9. Automatic Garbage Collection.
10. Portable.
11. Compiled & Interpretted
12. Inheritance
13. Polymorphism

Why is Java object-oriented ?
It supports Abstraction, Polymorphism, Encapsulation Inheritance, and 'all user-defined types must be object'.

Output of compiling → binary byte code.
JVM → Java Virtual Machine ⇒ container to run the program {Interpreter}
JVM makes it platform independent.

Inheritance ⇒ Person
Student Teacher
↓
Has own attr. as well

I

```
class HelloWorld {
    public static void main (String [] args)
{
    System.out.println ("hello World");
    }
}
```

Output: hello World

II

```
class no {
        public static void main (String[] arg)
{       int var = 23;
        int var1 = var % 2;
        if (var1 == 0){
            System.out.println ("even"); }
        else
            System.out.println ("odd");
}
}
```

~~CMD~~ → Open File in terminal.

① javac filename.java

② java filename.

12/12/23

I Rectangle area. Scanner:

```
import java.util.Scanner
Hello World
class RectangleArea{
    public static void main(String args[])
    { int a; float b; String s;
    Scanner in = new Scanner(System.in);
    System.out.println("~~You entered~~ a Enter
        String");
    s = in.nextline();
    System.out.println("You entered a steing"
        +s);
    System.out.println("Enter integer);
    a = in.nextInt();
    System.out.println("You entered an integer"
        + a);
    System.out.println("Enter float");
    b = in.nextFloat();
    System.out.println("You entered an float"
        +b); } }
```

II Rectangle Area.

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

```
int length, breadth;
length = Integer.parseInt(args[0]);
breadth = Integer.parseInt(args[1]);

int area = length * breadth;
System.out.println("Area = "+area);
  }
 }
```

III

Factorial

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

IV

```
import java.util.Scanner;
class palindrome {
    public static void main (String args[]){
        int n, t, rem, rev=0;
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a 5 digit
            number");
        n = sc.nextInt();
        t = n;
        while (t>0){
        rem = t%10;
        rev = rev*10 + rem;
        t = t/10;
        }
        if (rev==n){
            system.out.println("Palindrome");
        }
        else {
            System.out.println("Not palindrome");
        } } }
```

V

```
Array 1-D
class Auto Array{
    public Static void main (String args[])
    { int Mon-days = {31, 28, 30, 30, 31, 30, 31, 31, 30,
        31, 30, 31};
    System.out.println("April =" +mon-days + "days"); }}
```

Quadratic

```
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 coefficient
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, Enter non zero value
for a:");
Scanner s = new Scanner(System.in);
a = s.nextInt(); }
d = b*b - 4*a*c;
if (d==0) {
r1 = (-b) / (2*a);
System.out.println("Roots are real &
equal");
System.out.println("Root1 = Root2 ="+r1
}
```

else if (d > 0) {
r1 = ((-b) + (Math.sqrt(d)) / (double)(2*a);
r2 = ((-b) - (Math.sqrt(d)) / (double)(2*a);
System.out.println("Roots are real & distinct");
System.out.println("Root1 = " + r1 + "Root2 =" + r2); }

else if (d < 0)
{ System.out.println("Roots are imaginary");
r1 = (-b)/(2*a);
r2 = Math.sqrt(-d)/(2*a);
System.out.println("Root1 = " r1 + "+i" +r2);
System.out.println("Root2 = " r1 + "-i" +r2);
} } }

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

Output

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

II
Enter the coefficients of a, b, c
1 1 1
Roots are imaginary
Root 1 = 0 + i 0.866025
Root 2 = 0 - i 0.866025

III
Enter the coefficients of a, b, c
2 6 2
Roots are real and distinct
Root 1 = -0.38196 Root 2 = -2.618033

IV
Enter the coefficients of a, b, c
0 1 2
Not a quadratic equation. Enter a non zero value for a
b

Roots are imaginary.
Root 1 = 0 + i 0.571304.
Root 2 = 0 - i 0.571304.
Nidhi A - 1BM22CS177

VI) Sum of digits.

```
import java.util.Scanner;
class sumofdigits {
  public static void main (string args[])
  { long number, sum;
    Scanner sc = new Scanner (System.in);
    System.out.println("Enter a number");
    number = sc.nextLong();
  for(sum=0; number!=0; num=num/10)
    { sum = sum + number%10;}
  System.out.println("sum of digits:"
        + sum);
  } }
```

VII Conversion

```
class conversion {
public static void main (String args[])
{ byte b1, short s1 = 1000, s2;
  int i1 = 100000, i2;
```

```
long l1 = 1000000, l2;
char c = '+';
float f1 = 25.69f, f2;
double d1 = 536987.125, d2;
System.out.println("b+" + s1 + "
    i1 + "." + l1 + " " + c + " " + f1,
    "." + d1);
s2 = d
i2 = s1
l2 = i1;
System.out.println(s2 + " " + i2 +
    + l2);
}
}
```

13/12/23

LAB - 02

```
import java.util.Scanner;

class subject {
    int subjectMarks, credits, grade; }

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

    Student() {
        int i;
        subjects = new subject[9];
        for (i=0; i<8; i++){
        subjects[i] = new Subjects(); }
        s = new Scanner(System.in); }

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

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

        if(subjects[i].subjectMarks >= 40 &&
            subjects[i].subjectMarks <= 100){
        subjects[i].grade = calculateGrade(
                    subject[i].subjectMarks);}
        else
System.out.println("Invalid Marks");
        } System.out.println("Enter credits");
          subjects[i].credits = s.nextInt();}
    }

public int calculateGrade(int){
    if (marks >= 90){
            return 10;
    else if (moks >= 70 && marks <= 80)
            return 9;
    else if (marks >= 60 && marks <= 50)
            return 8;
    else
            return 7;
```

```
public void computeSGPA(){
    int totscore = 0;
    int totcred = 0;
    for (int i=0; i<8; i++){
    totscore += subjects[i].grade * subjects[i].cred;

    totcred += subjects[i].credits; }
    SGPA = (double) totscore / (double) totcred;
    }
}

public class Stud {
    public static void main(String args[]){
        Student s1 = new Student();
        s1.getStudentDetails();
        s1.getStudentMarks();
        s1.computeSGPA();

        System.out.println("Name: "+ s1.name);
        System.out.println("SGPA: "+ s1.SGPA);
    }
```

OUTPUT

Enter Name: Nidhi
Enter USN: 1BM22CS177
Enter Subject 1 marks : 100
Enter credits = 4

Enter Subjects 2 marks : 100
Enter credits = 4
Enter Subject 3 marks = 100
Enter credits = 3
Enter subject 4 marks = 100
Enter credits = 3
Enter subject 5 marks = 100
Enter credits = 3
Enter subject 6 marks = 100
Enter credits = 3
Enter subject 7 marks = 100
Enter credit = 1
Enter Subject 8 marks = 100
Enter credit = 1

Name = Nid
USN = 1BM22CS177
SGPA = 10

19/12/2023

Lab-03

Create a class book which contains 4 members name, author, price, num_pg. Include a constructor to set values of 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 program to create n books objects.

```
import java.util.Scanner;

class book {
    String name;
    String auth;
    int price;
    int pg_no;

    book (String name, String auth, int price, int
          pg_no) {
    this.name = name;
    this.auth = auth;
    this.price = price;
    this.pg_no = pg_no; }

    public String toString() {
```

```
String bookDetails = "Book name : " + this.name
+ "\n", "Author name : " + this.auth + "\n"
+ "Price " + this.price + "\n" );
return bookDetails;
}
}

public class Main {
public static void main ( String args[ ]
{ Scanner s = new Scanner(system.in);
System.out.println(" Enter the number of books
int n = s.nextInt();

Book[ ] books = new Book[n];

for(int i = 0; i < n; i++) {
System.out.println(" Enter details for book
+ "(i+1));
System.out.print(" Name: ");
String name = s.next();
System.out.print(" Price: ");
int price = s.nextInt();
System.out.print(" Number of pages: ");
int pg_num = s.nextInt();
```

books[i] = new Book (name, aut, price, pg-no);
{ System.out.println("Details of the books");
for (i=0; i<n; i++);
System.out.println("Book" + (i+1) + ":\n"
+ books[i].toString());
}
}
}

Output

Enter the number of books: 2
Book : 1
Name : Eragon
Author : Christopher
Pg-No : 2050
Price : 450

Book : 2
Name : Brisinger
Author : Pam
Pg-No : 2100
Price : 500

Details of the books:
Book 1 Name : Eragon
Author : Christopher
Pg-No : 2050

Price : 450

Book 2 : Name : Brisingr
Author : Ram
Pg-No : 2100
Price : 500

Nidhi
1BM22CS177

21-12-24

2/1/23

Lab-04

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

```
import java.util.Scanner;
class InputScanner {
    Scanner s = new Scanner(System.in);
    double getInput(String prompt){
        System.out.println(prompt);
        return s.nextDouble(); } }

abstract class Shape extends InputScanner{
    double side1, side2;
    abstract void area(){; }

class Rectangle extends Shape {
    Rectangle(){
        side1 = getInput("Enter rectangle length:");
        side2 = getInput("Enter rectangle breadth:");
    }
```

```
void area(){
    double area = side1 * side2;
    System.out.println("Area of rectangl=" area
    }}

class triangle extends Shape {
    triangle(){
    side1 = getInput("Enter base of triangle
    sede2 = getInput("Enter height of triangle
    }
    void area(){
    double area = 0.5 * side1 * side2;
    System.out.println("Area of triangle=" + ar
    }}

class circle extends shape{
    circle(){
    side1 = getInput("Enter radius of circle:");}}
    void area(){
    double area = Math.PI * side1 * side1;
    System.out.println("Area of circle=" + area);
    }}
```

```
class Main{
    public static void Main (){
        Rectangle rectangle = new Rectangle();
        Triangle Triangle = new triangle();
        circle Circle = new Circle ();

        rectangle.area();
        Triangle.area();
        Circle.area();}
        }
```

If negative values to be checked

```
class InputScanner{
  Scanner s = new Scanner (System.in);
  int getInput (String prompt){
    double input;
      do{
        System.out.println(prompt);
        input = s.nextDouble;
        if(input <0){
            System.out.println("Enter positive
                                values only");}
      } while(input< 0);
  return input; } }
```

Enter rectangle length: 12
Enter rectangle breadth: 10

Enter base of triangle: 5
Enter ~~base~~ height of triangle: 10

Enter radius of circle: 10

Area of rectangle = 120.000
Area of triangle = 25.000
Area of circle = 314.000

Rv 24/12/24

09/01/24

Lab - 05

Develop a java program to do create class Bank that maintains savings & current account. Accept deposit from customers and update the balance. Display the balance. Compute Interest

```
import java.util.Scanner;
class Account {
    String custName;
    long accNo;
    String accType;
    double balance;

public Account (String custName, long accNo,
    String accType, double balance){
        this.custName = custName;
        this.accNo = accNo;
        this.accType = accType;
        this.balance = balance; }

public void deposit( double amount) {
    balance += amount;
    System.out.println(" Deposit of $" + amount + "
    successful. Updated balance: $" + balance);
    }
```

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

public void withdraw(double amount) {
    System.out.println("Withdrawal not supported for this acc type"); }
}

class CurAccount extends Account {
    double MinBalance;
    double ServiceCharge;

    public CurAccount(String custname, long accNo, double balance,
        double MinBalance, double serviceCharge) {
        super(custName, accNo, "Current", balance);
        this.minBalance = MinBalance;
        this.serviceCharge = ServiceCharge; }

public void checkMinBalance() {
    if (balance < MinBalance) {
        balance -= serviceCharge;
        System.out.println("Minimum balance
            not maintained.
```

```
        Service charge of $" + serviceCharge + " imposed");
        displayBalance();
    }
}
@Override
public void withdraw(double amount){
    if (amount > balance){
        System.out.println("Insufficient funds");
    }
    else
    balance -= amount;
    System.out.println("$" + amount + " withdrawn
    Updated balance: $" + balance);
    checkMinBalance();
    }
}
}

class SavAccount extends Account {
    double interestRate;
public SavAccount(String custName, long accNo,
    double balance, double interestRate){
    super(custName, accNo, "savings", balance);
    this.interestRate = interestRate);
```

```
public void computeInterest() {
    double interest = balance * (intRate/100);
    balance += interest;
    System.out.println("Interest &
     deposit", interest);
    displayBalance;
  }
}
}
```

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

    System.out.println("Name,");
    String custName = s.nextLine();

    Syst.out.println("Enter initial amt");
    double IA = s.nextInt();

    System.out.println("Enter acc Type");
    Acc useraccDet = null;
```

Page No:
Date: / /

```
if (accountType = 1){
  Sys.printnt(" Enter minim balan for
     curreit   account); s.
  minbalance = s.nextDouble();
 Sys.println= (" Enter service charge");
  servicecharge = s.NextDoulle();
 userAccount = new SavAccount(custName,
    intialAmount, interestRate);}
else {
   System.out.println("Invalid tye Exit progr");
   s.close();
   return; }

int choice;
do {
  System.out.println(" Select \n")
  System.out.print("1. Deposit \n \2. Display
  Balance \n 3. compute Interest savings
   only \n4. Withdraw \n 5. Exit \n
     Enter Choices");
  choice = s.nextINt();
```

```
switch (choice) {
    case 1:
        System.out.println("Enter amount to
        deposit");
        double dep = s.nextDouble();
        userAcc.deposit(dep); break;
    case 2:
        userAcc.displayBalance();
        break;
    case 3:
        if(userAcc instanceof SavAccount){
        ((SavAccount)userAcc).computeInterest();
        else { out.
        System.println("Invalid opt for curr
            acc"); }; break;
    case 4: System.out.println("Enter
            amt to withdraw");
        double withdrawAmount = S.nextD
        userAccount.withdraw(withdrawAmount);
        break;
    case 5: System.out.println("Exiting
            the program"); break;
    default: System.out.println("Invalid
                choice"); }
} (while choice!=5);
```

```
scanner s.close();
    }
}
```

OUTPUT

```
Enter your name: Nidhi
Enter intial amount : 2000.
Select account type(1. Current 2 Savings)
    : 1
Enter min balance for current acc: 200
Enter service charge for  acc: 20

Select an option
1. Deposit
2 Display Balance
3 Compute Interest (Savings only)
4. (1, a. Withdraw
5. Exit
Enter choice : 1
Enter amount to deposit = $1000.

Select an option
1. {: Intenal choice again}
```

2
Account Balance: $3000

{Internal choice} : 3
Invalid option for current account

{Internal choice} : 4
Enter amount to withdraw : 100

{Enter choice} : 5
Exiting the program.

Enter your name: Nidhi
Enter intial amount : 2000
Type of account (1: current 2. Savings): 2
Enter interest rate : 5

Select an option:
1. Deposit
2. Display Balance
3. compute Interest
4. withdraw
5. Exit
Enter your choice : 3

Interest computed and deposited : $100.0
Account Balance : $2100.0.
USN - 1BM22LS177

9/9/2024

16/1/24

Lab - 06 ⇒ OUTPUT

1. Hello, World!
Hello
Java Programming
Java
ABCDE
Java
BCD

2. String 1: Hello,
String 2: Java!
Concated String: Hello, Java!
Length of concatenated String: 12

4 and 3. College Name = 'Welcome to Bmsce college'
Extracted Substring = Bmsce

5. Part 1 ⇒ Byte representation:
72 101 108 108 111 44 32 87 111
114 108 103 33
Original String: Hello, World!

Part 2 ⇒
Char array representation
J a v a P r o g r a m m
Original String: Java Programming

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

7. substring is matched

8. Hello, World! starts with Hello → true
and 9. Java Programming starts with Python → false
Hello World! ends with World! → true
Java Programming ends with C++ → false

10. Hello! equals Hello! → true
Hello! == Hello! → false

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

12. Sorted numbers (Descending order):
10 9 8 7 6 5 4 3 2 1

13. ~~Modified String: This is a test. Thiss was, too~~

14. Concatenated String: helloworld.

15. Thiss was a test. Thiss was, too
This was a test. Thiss was, too
This is a test. Thiss was, too
This is a test. This was, too
This is a test, This was, too

16. Original String: This is my college
Modified String: This is my commege.

17. Origadd String: ' Hello friends '
Trimmed String: 'Hello friends'.

18. After setLength(5): Hello
Character at index: 1: e
After setCharAt(1, 'a'): Hallo
Characters from index 0 and 4: Hallo
After append(): Hallo Appended!

After insert (6, 'Inserted'): Hallo Inserted Appended:
After reverse(): !dedneppA detereseruI allaH
After delet[illegible](5, 14): !deteresnI allaH
After deleteCharAt(0): deteresnI allaH
After replace(0, 4, 'Replaced'): ReplacedsnI allaH
Substring from index 3 to 7: laced.

19. Details about Eagle
Eagle flies high
Eagle has sharp cry

Details about Hawk
Hawk soars through air
Hawk emits high-pitched scream

20 Details about Circle
Area: 78.5398
Perimeter: 31.4159
Details about Triangle
Area: 6.0
Perimeter: 12.0

T.B

17. B Enter details for Studen 1:
Reg No: 1
Full Name: A
Sem: 1
CGPA: 9
Enter details for Student 2
Reg No: 2
Full Name: B
Sem: 1
CGPA: 8
Enter details for Student 3
Reg No: 3
Full Name: C
Sem: 1
CGPA 9.3
~~Sta~~ Enter details for Student 4
Reg No: 4
Full Name: D
Sem: 1
CGPA: 9.6
Enter details for Student 5
Reg No: 5
Full Name: E
Sem: 1
CGPA 8.7

Student record sorted by CGPA

Reg No: 4
Full Name: D
CGPA: 9.6

Reg No. 3
Full Name: C
CGPA: 9.3

Reg No: 1
Full Name: A
CGPA: 9

Reg No: 5
Full Name: E
CGPA: 8.7

Reg No. 2
Full Name: B
CGPA: 8.1

16/1/24

27/1/24

Lab 06

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

```
package CIE;
public class Internals extends Student {
public int imarks [] = new int [5];
public internals ( ) { }
public Internals (String name, int usn, int sem)
   super (name, usn, sem); } }
```

```
package SEE;
public class Externals extends CIE. student{
  public int emarks [ ] = new int [5];
}
```

```
import java.util.Scanner;
import CIE.student;
import CIE.Internals;
import SEE.Externals;

class Lab9Q6 {
    public static void main (String args[]){
        Scanner sc = new Scanner(System.in);
        String name;
        int usn, sem;
        System.out.println("Enter the no. of
            students");
        int n = sc.nextInt();
        Internals in[] = new Internals[n];
        Externals ex[] = new Externals[n];
        for(int i=0; i<n; i++){
        System.out.println("Enter name of student"
            +(i+1)+":");
        name = sc.nextLine();
        System.out.println("Enter usn "+(i+1)+":");
        usn = sc.nextInt();
        System.out.println("Enter sem"+(i+1)+":");
        sem = sc.nextInt();
        in[i] = new Internals(name, usn, sem);
        ex[i] = new External();
```

```
System.out.println("Enter 5 internals marks");
for(j=0; j<5; j++) {
    int[i].imarks[j] = sc.nextInt(); }
System.out.println("enter 5 external marks");
for (int j=0; i<5; j++){
    ex[j].emarks[j] = sc.nextInt(); }
}

for (int i=0; i<n; i++){
  System.out.println("Details of student" + (i+1) +
      ":");
  System.out.println("Name:" in[i].name + "\t"
    + "USN" + in[i].usn + "\t" + "sem:" + in[i].
  System.out.println("Internal marks:");
    for(int j=0; j<5; j++) {
      System.out.println("Subject" + (i+1) +
      in[i].imarks[j] + "\t"); }
      System.out.println("External mark:");
    for(int j=0; j<5; j++) {
      System.out.println("Subject" + (i+1) +
          + ex[i].emarks[j] + "\t");
      System.out.println("\nTotal marks:");
    for(int j=0; j<5; j++){
      System.out.println("Subject" + (i+1) +
      ( ) (INT.imarks[j] + ex[i].marks[j]
```

PAGE NO:
DATE: / /

OUTPUT

Enter no of students : 1
Enter name of student 1 : nidhi
Enter usn of 1 : 12345
Enter sem of 1 : 2

Enter the internal marks of student in 5 subjects :
45
34
43
28
48

Enter the external marks of student in 5 subjects
48
49
47
49
49

Details of student 1 :
Name : nidhi USN : 12345 sem : 2
Internal marks
Subject 1 : 45 Subject 2 : 34 Subject 3 : 43 Subject 4 : 28 Subject 5 : 48

External marks

Subject 1: 48 Subject 2: 49 Subject 3: 47

Subject 4: 49 Subject 5: 49.

Total marks

Subject 1: 93

Subject 2: 83

subject 3: 90

Subject 4: 77

Subject 5: 97.

Lab 07

Father - Son Age

```
import java.util.Scanner;
class WrongAge extends Exception{
    WrongAge(String error){
        System.out.println(error); }
}

class Father{
    int age;
    Father(int age) throws WrongAge{
        if(age<0)
        throw new WrongAge("Father's age cannot
        be negative");
        this.age = age; } }

class Son extends Father {
    int age;
    Son(int age, int s_age) throws WrongAge {
        super(age);
        if(s_age >= age){
        throw new WrongAge("Son's age cannot
        be greater than Father's Age");
        this.age = age; } }
```

```
class Main{
  public static void main(String args[]){
   Scanner sc = new Scanner(System.in);
   try {
    System.out.println("Enter Father's age:");
    int F_age = sc.nextInt();
   System.out.println("Enter Son's age:");
    int S_age = sc.nextInt();
   Son a = new Son(F_age, S_age);
   System.out.println("Father's age:" + F_age);
   System.out.println("Son's age:" + S_age);
  }
catch (WrongAge e){
System.out.println("Wrong age entered");}
catch(Exception ee){
 System.out.println("Unexpted error:" + ee);}
  }
}
```

OUTPUT

Enter the Father's Age 53
Enter the Son's Age : 15
Father's age : 53
Son's age : 15

Enter Father's age : 15
Enter Son's age : 23
~~Wrong~~ Son's age cannot be greater than Father's age
Wrong age entered

Enter the Father's age : -15
Enter the Son's age = 15
Father's age cannot be negative
Wrong Age entered.
NITIE
IBM22C5177

30/12/2022

Lab - 08

```
class DisplayMessage extends Threads {
    private final String message;
    private final long interval;

    DisplayMessageThread (String message, long interval)
    { 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.currentThread()
            getName() + "interrupted"); }
    } }

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

        DisplayMessageThread thread1 = new DisplayMessa
            thread ("BMS College of Engineering", 10000);
        DisplayMessageThread thread2 = new DisplayMessag thread
            ("CSE", 2000);
```

```
thread1.setName("Thread1");
thread2.setName("Thread2");
thread1.start();
thread2.start();
try { Thread.sleep(30000); }
catch (InterruptedException e) {
        System.out.println("Main thread interrupted");
        }
thread1.interrupt();
thread2.interrupt();
System.out.println("Main thread Exiting"); }}
```

OUTPUT

```
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
Main thread exiting
Thread 2 interrupted.
Thread 1 interrupted.
```

Lab-10

IPC

```
class Q {
    int n;
    boolean valueSet = false;
    synchronized int get() {
        while (!valueSet)
        try {
            System.out.println("\n Consumer waiting\n");
        wait();} catch(InterruptedException e){
            System.out.println("Interruption caught");
        System.out.println("Got " + n);
        valueSet = false;
        System.out.println("\n Intimate Producer");
        notify();
        return n; }
    synchronized void put(int n) {
        while(valueSet) {
        try {
            System.out.println("\n Producers waiting\n");
            wait(); } catch (InterruptedException e)
                System.out.println("Interrupted Exception
                    caught"); }
        this.n = n;
        valueSet = true;
```

```
        System.out.println("Put: " + n);
        System.out.println("\n Intimate Consumer\n");
        notify(); } }

class Prodeserce implements Runnable {
    Q q;
    Producer (Q q) {
        this.q = q;
        new Thread (this, "Producer").start(); }
    public void run(){
        int i = 0;
        while (i<=15){ q.put(i++); }}}

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

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

Output

```
Put : 0
Intimate Consumer
Producer waiting
Got : 0
Intimate Producer
Put : 1
Intimate Consumer
Producer waiting
Consumed : 0
Got : 1
Intimate Producer
Consumed : 1
Put : 2
Intimate Consumer
Producer waiting
Got : 2
```

6/7/2021

13/2/24

10 - Deadlock

```
class A {
    synchronized void foo(B b) {
    String name = Thread.currentThread().getName();
    System.out.println(name + "entered A.foo");
try {
  Thread.sleep(5000); }
catch (Exception e){
    System.out.println("A interrupted"); }
System.out.println("trying to call B.last()", name);
b.last(); }
void last() {
    System.out.println("Inside A.last"); } }

class B {
  synchronised void bar(A a) {
    String name = Thread.currentThread().getName()
  System.out.println(name + "entered B.name");
try {
    Thread.sleep(1000); }
catch (Exception e) {
    System.out.println(name + "B. interrupted"
} System.out.println(name + "trying to call A.last()");
a.last(); }
void last(){
```

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

class Deadlock implement Runnable {
    A a = new A();
    B b = new B();
    Deadlock() {
        Thread.currentThread().setName("MainThread");
        Thread t = new Thread(this, "Racing Method");
        t.start();
        a.foo(b);
        System.out.println("Back in other thread");
    }
    public void run() {
        b.bar(a);
        System.out.println("Back in other thread");
    }
    public static void main(String args[]) {
        new Deadlock(); }
}
```

OUTPUT

```
MainThread entered a.foo
RacingThread entered b.foo
MainThread trying to calling B.last()
```

Inside A.last
Back in main thread
RacingThread trying to call A.last().
Inside A.last
Back in other thread

Nidhi
1BM22CS177

13/2/2024

20/02/24

Lab-09

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

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

        JLabel jlab = new JLabel("Enter the divident
        and divider");
        JTextField ajtf = new JTextField(8);
        JTextField bjtf = new JTextField(8);
        JButton button = new JButton("Calculate");

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

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

ActionListener l = new ActionListener();
public void actionPerformed(ActionEvent
    et) {
    try {
        int a = Integer.parseInt(bjtf.getText());
        int ans = a/b;
        alab.setText("\n A =" + a);
        lblab.setText(" \n B=" + b);
        anslab.setText("\n Ans = ", + ans); }
catch (No Format Exception e) {
        alab.setText(" ");
        lblab.setText(" ");
        anslab.setText(" ");
        err.setText("Enter only Integers"); }
```

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

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

OUTPUT

Divider App

Enter the divident and divider :

[200] [2]

[Calculate] A = 200 B = 2 Ans = 100.

20.02.24

Report on AWT Functions

PAGE NO:
DATE: / /

JFrame: It is a class in Java that is part of the swing library, which is used for creating graphical user interface in Java application

setSize(): is a method which is used with component such as JFrame, JPanel etc to set their size

setLayout(): It is a method which is used to set the manager for a container such as JFrame & any other container component

setDefaultCloseOperation(): It is a method which is used to specify default close operation for a JFrame

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

JTextField: It is a class that provides a ~~text~~ input Field in a GUI

Add ActionListener: add ActionListener is a method that is used to register an ActionListener for a component, typically for a button or any other interactive component that generate action events.

setText: It is method used to set text content of text based components

Thu
20.02.24

LAB-01 QUADRATIC EQUATION

Question

Develop a Java program that prints all real solutions to the quadratic equation ax 2 +bx+c = 0. Read in a, b,

c and use the quadratic formula. If the discriminate b 2 -4ac is negative, display a message stating that there are no real solutions

Code:

```
import java.util.Scanner;
class Quadratic
{
int a, b, c;
double r1, r2, d;
void getd()
{
Scanner s = new Scanner(System.in);
System.out.println("Enter the coefficients of a,b,c");
a = s.nextInt();
b = s.nextInt();
c = s.nextInt();
}
void compute()
{
while(a==0)
{
System.out.println("Not a quadratic equation");
System.out.println("Enter a non zero value for a:");
Scanner s = new Scanner(System.in);
a = s.nextInt();
}
d = b*b-4*a*c;
if(d==0)
{

r1 = (-b)/(2*a);
System.out.println("Roots are real and equal");
System.out.println("Roo1 = Root2 = " + r1);
}
else if(d&gt;0)
```

```
{
r1 = ((-b)+(Math.sqrt(d)))/(double)(2*a);
r2 = ((-b)-(Math.sqrt(d)))/(double)(2*a);
System.out.println("Roots are real and distinct");
System.out.println("Roo1 = " + r1 + " Root2 = " + r2);
}
else if(d&lt;0)
{
System.out.println("Roots are imaginary");
r1 = (-b)/(2*a);
r2 = Math.sqrt(-d)/(2*a);
System.out.println("Root1 = " + r1 + " + i"+r2);
System.out.println("Root1 = " + r1 + " - i"+r2);
}

}
}

class QuadraticMain
{
public static void main(String args[])
{
Quadratic q = new Quadratic();
q.getd();

q.compute();
}
}
```

LAB-02- STUDENT SGPA CALCULATION

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.

Code:

```
import java.util.Scanner;


class subject{
int subjectMarks, credits, grade;}
class Student {
    String name;
    String usn;
    double SGPA;
    Scanner s;
    subject subjects[];


Student()
{
int i;
subjects = new subject[9];
for(i=0;i<8;i++)
subjects[i] = new subject();
s = new Scanner(System.in);
}


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

```
System.out.println("Enter Student USN:");
usn=s.nextLine();}

public void getMarks(){
int i;
for(i=0;i<8;i++){
System.out.println("Enter marks of subject"+(i+1)+":");
subjects[i].subjectMarks= s.nextInt();
if(subjects[i].subjectMarks>=40&&subjects[i].subjectMarks<=100){
subjects[i].grade=calculateGrade(subjects[i].subjectMarks);}
else{
System.out.println("Invalid Marks. Marks should be between 40 and 100");}
System.out.println("enter credits:");
subjects[i].credits=s.nextInt();
}
}

public int calculateGrade(int marks){
if (marks>=90)
return 10;
else if(marks>=70&&marks<=80)
return 9;
else if(marks>=60&&marks<=70)
return 8;
else if(marks>=50&&marks<=60)
return 7;
else
return 6;
}
```

```
public void computeSGPA() {
        int totalscore = 0;
        int totalcred = 0;

        for (int i = 0; i < 8; i++) {
            totalscore += subjects[i].grade * subjects[i].credits;
            totalcred += subjects[i].credits;
        }
        SGPA = (double) totalscore / (double) totalcred;
    }
}

public class Stud{
public static void main(String args[]){
Student s1=new Student();

s1.getStudentDetails();

s1.getMarks();
s1.computeSGPA();

System.out.println("Student name:"+s1.name);
System.out.println("Student usn:"+s1.usn);
System.out.println("Student sgpa:"+s1.SGPA);}
}
```

LAB-03 BOOK PROGRAM

Question:

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.

CODE:

```
import java.util.Scanner;


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

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

    // toString method to display book details
    public String toString() {
        String bookDetails = "Book name: " + this.name + "\n"
                + "Author name: " + this.author + "\n"
                + "Price: " + this.price + "\n"
                + "Number of pages: " + this.numPages + "\n";
        return bookDetails;
```

```
    }
}

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

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

        // Define array of Book objects
        Book[] books = new Book[n];

        // Input details for each book
        for (int i = 0; i < n; i++) {
            System.out.println("Enter details for Book " + (i + 1));
            System.out.print("Name: ");
            String name = s.next();
            System.out.print("Author: ");
            String author = s.next();
            System.out.print("Price: ");
            int price = s.nextInt();
            System.out.print("Number of pages: ");
            int numPages = s.nextInt();

            // Create a new Book object with the entered details
            books[i] = new Book(name, author, price, numPages);
        }
```

```
        // Display details of each book
        System.out.println("\nDetails of the books:");
        for (int i = 0; i < n; i++) {
            System.out.println("Book " + (i + 1) + ":\n" + books[i].toString());
        }
    }
}
```

LAB-04 SHAPE PROGRAM

Question:

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

CODE:

```
import java.util.Scanner;
class InputScanner{
Scanner s= new Scanner(System.in);
double getInput(String prompt){
System.out.println(prompt);
return s.nextDouble();}}
abstract class Shape extends InputScanner{
double side1, side2;
abstract void area();}

class Rectangle extends Shape{
Rectangle(){
side1=getInput("Enter length of rectangle:");
side2=getInput("Enter breadth of rectangle:");}
void area(){
double area=side1*side2;
System.out.println("Area of the Rectangle ="+area);}}
class triangle extends Shape{
    triangle(){
    side1=getInput("Enter base of the triangle:");
    side2=getInput("Enter height of the triangle:");}
     void area(){
```

```
        double area=side1*side2/2;
System.out.println("Area of the triangle="+area);}}
class circle extends Shape{
circle(){
side1=getInput("Enter the radius of the circle:");}
void area(){
double area=Math.PI*side1*side1;
System.out.println("Area of the circle="+area);}}

class Main{
public static void main(String args[]){
Rectangle rectangle= new Rectangle();
triangle Triangle=new triangle();
circle Circle=new circle();

rectangle.area();
Triangle.area();
Circle.area();

}
}
```

LAB -05 BANK ACCOUNT(abstract classes)

Question

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.

CODE:

```
import java.util.Scanner;


class Account {
    String customerName;
    long accountNumber;
    String accountType;
    double balance;

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

    public void deposit(double amount) {
        balance += amount;
        System.out.println("Deposit of $" + amount + " successful. Updated balance: $" +
 balance);
    }

    public void displayBalance() {
        System.out.println("Account Balance: $" + balance);
```

```
    }
    public void withdraw(double amount) {
        System.out.println("Withdrawal not supported for this account type.");
    }
}

class CurAccount extends Account {
    double minBalance;
    double serviceCharge;

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

    public void checkMinBalance() {
        if (balance < minBalance) {
            balance -= serviceCharge;
            System.out.println("Minimum balance not maintained. Service charge of $" +
 serviceCharge + " imposed.");
            displayBalance();
        }
    }
    @Override
    public void withdraw(double amount) {
        if (amount > balance) {
            System.out.println("Insufficient funds. Withdrawal failed.");
        } else {
            balance -= amount;
```

```
            System.out.println("Withdrawal of $" + amount + " successful. Updated balance: $" +
balance);
            checkMinBalance();
        }
    }
}

class SavAccount extends Account {
    double interestRate;

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

    public void computeInterest() {
        double interest = balance * (interestRate / 100);
        balance += interest;
        System.out.println("Interest computed and deposited: $" + interest);
        displayBalance();
    }
    @Override
    public void withdraw(double amount) {
        if (amount > balance) {
            System.out.println("Insufficient funds. Withdrawal failed.");
        } else {
            balance -= amount;
            System.out.println("Withdrawal of $" + amount + " successful. Updated balance: $" +
balance);
        }
```

```
    }
}

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

        System.out.print("Enter your name: ");
        String customerName = scanner.nextLine();

        System.out.print("Enter initial amount: ");
        double initialAmount = scanner.nextDouble();

        System.out.print("Select account type (1. Current, 2. Savings): ");
        int accountTypeChoice = scanner.nextInt();

        Account userAccount = null;  // Declare userAccount outside the if-else blocks

        if (accountTypeChoice == 1) {
            System.out.print("Enter minimum balance for the current account: ");
            double minBalance = scanner.nextDouble();
            System.out.print("Enter service charge for the current account: ");
            double serviceCharge = scanner.nextDouble();

            userAccount = new CurAccount(customerName, System.currentTimeMillis(),
initialAmount, minBalance, serviceCharge);
        } else if (accountTypeChoice == 2) {
            System.out.print("Enter interest rate for the savings account: ");
            double interestRate = scanner.nextDouble();

            userAccount = new SavAccount(customerName, System.currentTimeMillis(),
```

```
initialAmount, interestRate);
        } else {
            System.out.println("Invalid account type choice. Exiting the program.");
            scanner.close();
            return;
        }

        int choice;
        do {
             System.out.println("\nSelect an option:");
            System.out.println("1. Deposit");
            System.out.println("2. Display Balance");
            System.out.println("3. Compute Interest (Savings Account only)");
            System.out.println("4. Withdraw");
            System.out.println("5. Exit");
            System.out.print("Enter your choice: ");
            choice = scanner.nextInt();

            switch (choice) {
                case 1:
                    System.out.print("Enter amount to deposit: ");
                    double depositAmount = scanner.nextDouble();
                    userAccount.deposit(depositAmount);
                    break;
                case 2:
                    userAccount.displayBalance();
                    break;
                case 3:
                    if (userAccount instanceof SavAccount) {
                        ((SavAccount) userAccount).computeInterest();
                    } else {
```

```
                System.out.println("Invalid option for current account.");
            }
            break;
        case 4:
            System.out.print("Enter amount to withdraw: ");
            double withdrawAmount = scanner.nextDouble();
            userAccount.withdraw(withdrawAmount);
            break;
        case 5:
            System.out.println("Exiting the program. Thank you!");
            break;
        default:
            System.out.println("Invalid choice. Please enter a valid option.");
      }
    } while (choice != 5);

    scanner.close();
  }
}
```

LAB-06 PACKAGES

Question:

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

CODE:

```
package SEE;
public class External extends CIE.Student{
	public int emarks[]=new int[5];
}

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

package CIE;
public class Internals extends CIE.Student{
	public int imarks[]=new int[5];
	public Internals(){}
	public Internals(String name,int usn,int sem){
		super(name,usn,sem);
	}
}
```

```
import java.util.Scanner;
import CIE.Student;
import CIE.Internals;
import SEE.External;
class LabQ6{
	public static void main(String args[]){
		Scanner sc=new Scanner(System.in);
		String name;int usn,sem;
		System.out.println("Enter the number of students:");
		int n=sc.nextInt();
		Internals in[]=new Internals[n];
		External ex[]=new External[n];
		for(int i=0;i<n;i++){
			System.out.println("Enter the name of the student "+(i+1)+":");
			name=sc.next();
			System.out.println("Enter the USN of the student "+(i+1)+":");
			usn=sc.nextInt();
			System.out.println("Enter the sem of the student "+(i+1)+":");
			sem=sc.nextInt();
			in[i]=new Internals(name,usn,sem);
			ex[i]=new External();
			System.out.println("Enter the internal marks of the student in 5
			subjects:");
			for(int j=0;j<5;j++)
				in[i].imarks[j]=sc.nextInt();
			System.out.println("Enter the external marks of the student in 5
			subjects:");
			for(int j=0;j<5;j++)
				ex[i].emarks[j]=sc.nextInt();
		}
```

```
        for(int i=0;i<n;i++){
                System.out.println("Details of student "+(i+1)+":");
                System.out.println("Name:"+in[i].name+"\t"+"USN:"+in[i].usn+"\t"+"Sem:"+i
                n[i].sem);
                System.out.println("Internal marks:");
                for(int j=0;j<5;j++)
                        System.out.print("Subject "+(i+1)+":"+in[i].imarks[j]+"\t");
                System.out.println("External marks:");
                for(int j=0;j<5;j++)
                        System.out.print("Subject "+(j+1)+":"+ex[i].emarks[j]+"\t");
                System.out.println("\nTotal marks:");
                for(int j=0;j<5;j++)
                        System.out.println("Subject"+(j+1)+":"+(in[i].imarks[j]+ex[i].emarks[j])
                );
        }
}}
```

LAB -07 EXCEPTIONS (FATHER-SON)

Question:

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

CODE:

```
import java.util.Scanner;
class WrongAge extends Exception{
	WrongAge(String error){
		System.out.println(error);
	}
}

class Father{
	int age;
	Father(int age) throws WrongAge{
		if(age<0)
			throw new WrongAge("Father's age cannot be negative");
		this.age=age;
	}
}

class Son extends Father{
	int age;
	Son(int age,int s_age) throws WrongAge{
		super(age);
		if(s_age>=age)
			throw new WrongAge("Son'age cannot be greater than Father's age");
		this.age=s_age;
```

```
	}
}

class LabQ7{
	public static void main(String args[]){
		Scanner sc=new Scanner(System.in);
		try{
			System.out.println("Enter the Father's age:");
			int f_age=sc.nextInt();
			System.out.println("Enter the Son's age:");
			int s_age=sc.nextInt();
			Son a=new Son(f_age,s_age);
			System.out.println("Father's age:"+f_age);
			System.out.println("Son's age:"+s_age);
		}
		catch(WrongAge e){
			System.out.println("Wrong Age entered");}
		catch(Exception ee){
			System.out.println("Unexpected error :"+ee);}
		}
}
```

LAB-08 THREADS

Question:

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.

CODE:

```
class DisplayThread extends Thread {
    private String message;
    private int interval;
    private boolean running = true;
    public DisplayThread(String message, int interval) {
        this.message = message;
        this.interval = interval;
    }
    public void run() {
        while (running) {
            System.out.println(message);
            try {
                Thread.sleep(interval);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
    public void stopThread() {
        running = false;
    }
}
public class ThreadExample {
    public static void main(String[] args) {
        DisplayThread bmsThread = new DisplayThread("BMS College of Engineering",
```

```
10000);
        DisplayThread cseThread = new DisplayThread("CSE", 2000);
        bmsThread.start();
        cseThread.start();
        System.out.println("Press Enter to stop the threads...");
        try {
            System.in.read();
        } catch (Exception e) {
            e.printStackTrace();
        }
        bmsThread.stopThread();
        cseThread.stopThread();
    }
}
```

LAB-09 CREATE USER INTERFACE

Question:

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.

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(l);

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

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

                blab.setText("");
                anslab.setText("");
                err.setText("B should be NON zero!");
            }
        }
    });
    jfrm.setVisible(true);
  }
  public static void main(String args[]){
    // create frame on event dispatching thread
    SwingUtilities.invokeLater(new Runnable(){
      public void run(){
        new SwingDemo();
      }
    });
  }
}
```

LAB-10 IPC AND DEADLOCK

IPC 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 {
        Q q;
        Producer(Q q) {
                this.q = q;
                new Thread(this, "Producer").start();
        }
        public void run() {
                int i = 0;
                while(i<15) {
                        q.put(i++);
                }
        }
}
class Consumer implements Runnable {
        Q q;
        Consumer(Q q) {
                this.q = q;
                new Thread(this, "Consumer").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.");
        }
}
```

DEADLOCK 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();
B b = new B();
Deadlock() {
Thread.currentThread().setName("MainThread");
```

```
Thread t = new Thread(this,"RacingThread");
t.start();
a.foo(b);
System.out.println("Back in main
thread");
}
public void run() {
b.bar(a);
System.out.println("Back in other thread");
        }
public static void main(String args[]) {
        new Deadlock();
}
}
```