B.M.S COLLEGE OF ENGINEERING BENGALURU

Autonomous Institute, Affiliated to VTU

OBJECT ORIENTED JAVA PROGRAMMING

Bachelor of Engineering
in
Computer Science and Engineering

Submitted by:

Name : CDT. PRATIK JANA

USN:1BM22CS356

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

Lab 1:

```
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 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 eqn");
            System.out.println("Enter a non-zero value for a");
            Scanner S= new Scanner(System.in)
            a= S.nextInt();
      }

      d= b*b-4*a*c
      if (d=0)
          { r1=(-b)/(2*a);
            System.out.println("Roots are real and equal");
            System.out.println("Root1 and root 2= "+r1");
          }
else if (d>0){
r1= ((-b) +(math.sqrt(d)))/(double)(2*a)
```

Date

Expt. No.

Page No.

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

    }

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

Class Quadratic Main{
    Public static void main (String args[]){
        Quadratic q= new Quadratic();
        System.out.println ("USN- 1BM22EC172 ,Name - Prabir");

        q.getd();
        q.compute();
            } }
```

output

Enter the coefficient a, b, and c,

1

-3

2

roots are real and distinct

Root1 = 2.0 and Root2 = 1.0

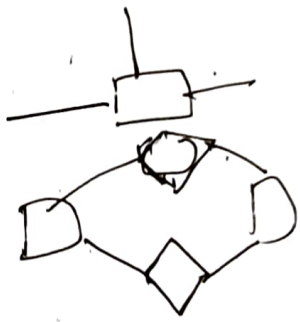
Teacher's Signature :

Develop a java programs to create a class student with members USN, Name, array marks. Include methods to accept and display details and method to calculate SGPA of Student

```
package LABPrograms;
import java.util.*;
class Subject{
    int Subjectmarks;
    int credits;
    int grade;
    }
class StudentClass1{
    string name;
    string USN;
    double SGPA;
    Scanner S = new Scanner(System.in);
    Subject[] subject;

    StudentClass1(){
        subject = new Subject[9];
        for (int i=0;i<9;i++) Subject[i] = new subject();
        }

    void getStudentDetails(){
        System.out.print ("Enter Student name:");
        name = S.nextline();
        System.out.print(" Enter Student USN");
        usn = S.nextline();
        }
```



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

  if (Subject[i].subjectMarks >=90){
     Subject[i].grade = 10; }
  else if (Subject[i].subjectMarks >= 80){
     Subject[i].grade = 9; }
```

```
>void computeSGPA(){
    double totalCredits=0;
    double WeightedSum=0;

    for(int i=0; i<9; i++){
    totalcredits= totalcredits + subject[i].credits;
    WeighedSum = WeightedSum + (subject[i].credits* object
                                                    grade
    }
    SGPA= WeightedSum / totalcredits;
}

>void displayResult(){
    System.out.println("Student Details:");
    System.out.println("Name" +name);
    System.out.println("USN:" +USN);
    System.out.println("SGPA:" +SGPA);
}

class studentClass{
    public static void main(String[] args){
        StudentClass1 S1 = new StudentClass1();
        S1.getStudentDetails()
        S1.getMarks();
        S1.computeSGPA();
        S1.displayResult();  }}
```

Enter Student name : PRATIK JANA
Enter Student USN : 1BM22EC172

Enter details of subject 1
Enter marks 98
Enter credits 4

89
4

89
3

89
3

89
3

99
1

99
1

99
1

87
3

Student details:

Name PRATIK JANA

USN 1BM22EC172

SGPA 9.304347826086957

Lab3 program:-

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 details of the objects. Include a toString() method that could display the complete details of the book. Develop a java program to create n books objects

```
import java.util.*;

class Books {
  string name, author;
  int price, numpages;
  Books( String name, String author, int price, int num pages)
  { this.name = name;
    this.author = author;
    this.price = price;
    this.numpages = numpages;

  String toString() {

      String author, name, price, numpages;
      name = "This BOOK is" + this.name + "\n";
      author = "Author is " + this.author + "\n");
      price = "price =" + this.price + "\n";
      numpages = "No of pages" + this.numpages + "\n"; }

    }
```

Expt. No. ---------- Page No. ----------

```
Public Class Main{
    Publis Static Void main(String[] args){
        System.out.println("USN-1BM22EC172 , Name - Prahrj");
        Scanner S= new Scanner(system.in);
        Sring name, author;
        int n, numpages, price;
        System.out.println("Enter the no of Books");
        n= S.nextInt();
        Books arr[];
        arr = new Books [n];
        for(int i=0, i<n; i++){
         System.out.println("Enter the name of the BOOK")
         name = S.nextInt();
         System.out.println("Enter the author of the Book")
         author= S.nexInt();
         System.out.println("Enter the price of the book")
         price = S.nextInt();
         System.out.println("Enter the no of pages of
                                the book")
         Price = S.nextInt();
         System.out.println("\n
         arr[i] = new Book(name, author price, numpages)
            }
        for(int i=0; i<n; i++){
            System.out.println(arr[i].toString());
            }
        }
    }
```

Teacher's Signature : _____

Output

USN: 1BM22EC172 Name: Pratik Jana
Enter no of BOOKS 2

- Enter name of Book
  Java
- Enter the author of BOOK
  500 Herbert
- Enter the price of book
  800
- Enter the no of pages of Book
  300
- Enter Name of the Book
  C
- Enter author of Book
  Ritchie
- Enter the price of book
  400
- Enter the no of pages of the book
  400

This Book is Java
Author is Herbert
Price = 500
No of pages = 300

This Book is C
Author is Ritchie
Price = 400
No of pages 200

Date

Expt. No. Lab4

Page No.

Q: Develop a java program to create 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 class extends the class shape.

Program:

```
import java.util.Scanner;
class InputScanner{
    Scanner S;
  InputScanner(){
        S= new Scanner(System.in);}

abstract class shape Extends Input Scanner{
  double a;
  double b;
  abstract void getdetails();
  abstract void printArea();}
  class Rectangle extends Shape{
      void getDetails(){
        System.out.println("Enter lenght and breadth");
        a= S.nextDouble();
        b= S.nextDouble();}
```

Teacher's Signature : _____

```
void printArea() {
        System.out.println("Area of Rectangle = " + (a * b)) }
    } }

Class Triangle Extends Shape {
    void getDetails() {
            System.out.println("Enter a and b");
            a = S.nextDouble();
            b = S.nextDouble(); }
    void printArea() {
            System.out.println("Area of triangle = " + (a*b)/2);
    } }

class circle Extends Shape {
    Void getDetails() {
        System.out.println("Enter Radius");
        a = S.nextDouble(); }

    void PrintArea() {
            System.out.println("Area of circle = " + (3.14 * a * a);
    } }
```

Date

Expt. No.

Page No.

```
class Abshact Main){

Public Static void mains(string[] args){
    Rectangle r = new Rectangle ();
    Triangle t = new Triangle. ();
    circle c = new circle ();
    r = get details();
    r = printArea.();
    t = getDetails();
    t = printArea();
    t. get Details();
    c. get printArea();
    System out.println ("Name - Prabik Jana") ; } }
```

output-

Enter length and breadth
5
8
Area of Rectangle - 40.0
Enter a & b
3
8
Area of Triangle = 12.0
Enter radius
2
Area of circle : 12.56
~~Name~~ = ~~Sourabh~~
Name: Prathk Jana.

Expt. No.

Date

Page No.

Lab 5: Strings and generics

1. Demonstrate various string constructor

abc
Hello

2. concatenated string, string' lenght, string literal

Lenght of S1 = 5
concatenated String: Hello World.

3. Demonstrate 'toStrings()':

This is a DemotoString object.

4. using 'getchars()', Extract "BMSCE" from "WELCOME to BMSCE College

Extracted String: BMSCE

Demonstrate 'getBytes()', 'tocharArray()';

5. Byte array: [74, 97, 118, 87, 32, 97, 105, 115, 32, 102, 117, 110, 33]
char array: [J, a, v, a, , i, s, , f, u, n, !]

Teacher's Signature: _____

6) check the following output and write the java programs.

(Bmsce equals Bmsce → true
Bmsce equals college → false
Bmsce equals BMSCE → false
Bmsce equalsIgnoreCase BMSCE → true)

7) using 'regionmatches()', find the substring 'BMSCE College':

substring is not matched

8) ~~using 'regionMatches()', find substring "BMSCE COLLEGE";~~
Demonstrate endswith()

true
false
true
false

9) Demonstrate ~~start~~ end with()

10) Demonstrate a java program to show o/p for 'equals()' versus '==':

true
false
true

Date

Expt. No. Page No.

Lab-6

Create a package CIE which has two classes Students and Internals. The class Student has member USN, name, sem. The class Internals derived from USN, Name, Sem. The class Internals members like USN, Name, Sem. The class Internals derived from the ...

cie/student.java

```
package Cie; import java.util.Scanner;
public ~~Util~~ class Student { Scanner Sc = new Scanner
Public String name, USN;                    (System.in);
Public Int Sem;
public void setDetails (){
    name = Sc.nextLine();
    USN = Sc.nextLine();
    Sem = Sc.nextInt();
}
public void getdetails(){
    System.out.println("Name:" + name);
    System.out.println("USN:" + USN);
    System.out.println("USN:" + USN);
```

Teacher's Signature : _____

(21)

Packages: Interfaces: ⊕ packao

cie/Internals.java

```
package cie;
import java.util.Scanner;
public class Internals extends student {
  public int internalMarks[] = new int[5];
  Scanner sc = new Scanner(System.in);

  public void setCie(){
    for(int i=0; i<5; i++){

      System.out.println("Enter marks of" + (i+1));
      internalMarks[i] = sc.nextInt();

    }}
```

See/External

```
package see;
import java.util.scanner;
import cie.Internals;
public
  class External extends Internals{

      public int seeMarks[] = new int[5];
      scanner sc = new scanner(system.in);

public void setsee(){
    for(int i=0; i<5; i++){
          System.out.println("Enter see marks of" (i+1));
      seeMarks[i] = sc.nextInt(); }}
```

Date

Expt. No. Page No.

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

DemoMain.java

```
import sel.External;
public class DemoMain{
    public static void main(String[] args){

        External obj = new external();
        obj.getdetails();
        obj.setdetails();
        obj.setcie();
        obj.getsee();
        obj.computefinal();
        obj.displayMarks();

}}
```

Teacher's Signature :

Output

Pratik Jana
1BM22EC172

Enter the cie marks of 5 subjects
50
40
35
40
50

Enter the see marks of 5 subjects

100
95
65
85
90

Subject 1 = 100
subject2 = 87
subject3 = 67
subject4 = 82
subject5 = 95

30/1/24

Date

Expt. No. ..... Page No.

Lab 7:-

Write a program that demonstrates handling of Exception in inheritance tree. Create a base class called father and derived class called Son which extends base class. In father class Implement a constructor which takes the age and throws the Exception wrongAge() when the input age < 0. In Son class, implement a constructor that cases both father & sons and throws an Exception.

```
import java.util.*;
class wrongAge Extends Exception {
        public wrongage(String A){
                super(A);
        }}

class Father{
int FatherAge;
Scanner SC = new Scanner (System.in);
Public Void Validage().throws wrongAge{
 System.out.println("Enter father age");
 father Age = Sc.nextInt();
 if (father Agge <=0){
      throw new wrongAge ("Invalid father age");
}}
}
```

Teacher's Signature : _____

```
class son Extends Father {
int sonAge;
scanner sc = new Scanner (system.in);
public void validAge() throws WrongAge {
        system.out.println ("Enter son age ");
        son Age = sc.nextInt();
        super.validAge();

if (sonAge >= fatherAge){
    throw new WrongAge ("son's age cant be greater");

    }

else if (sonAge < 0){
        throw new wrongAge ("Invalid son age")

    }}

public class MyMain {
    public static void main (String[] args) {
      son obj = new Son();
      try {
         obj.validAge();
      }
    catch (WrongAge e) {
        system.out.println ("Exception") }

                    }
                }
```

Date

Expt. No.

Page No.

Output

Enter's Son Age
30
Enter father Age
20

Exception Sons age can't be greater than father age

Enter son Age
7
Enter father age
20

Expt. No. Lab 8 Page No.

Date

a) Multithreading program:

```
package javaLabPrograms;

class name extends Thread {
    public static void run() { while(true) {
        System.out.println("BMS College of Engineering");
        try{
            Thread.sleep(millis:10000);
        }catch{ InterruptedException e) {
            System.out.println("interrupted")
            }
        }
    }
}

class nameThread extends Thread {
    public void run(){
    while(true){
        System.out.println("CSE");
    try {
        thread.sleep(millis:2000);
    } catch(InterruptedException e){
        System.out.println("interrupted");
    }
}
}
```

Teacher's Signature : _____

```
public class threads{
    public static void main(String[] args){
        System.out.println("Java");
        CNameThrea t1 = new CNameThread();
        DNameThread t2 = new DNameThread();

            t1.start();
            t2.start();
             }
          }
```

output

Java
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE

Date .....

Expt. No. .....

Page No. .....

Lab-9

open end Question

JFrame
setsize
setlayout
JLabel
JTextfield
addframe
addActionistrue

29/2/24

Teacher's Signature : _____

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

class SwingDemo{}

    SwingDemo(){

        JFrame jfrm = new JFrame("Divider App");
        jfrm.setSize(275,150);
        jfrm.setLayout(new FlowLayout());
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        JLabel jlab = new JLabel("Enter the dividend
                                  and divident")
        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(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");
    }
    aftf.addActionListener(l);
    bftf.addActionListener(l);

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

        alab.setText("");
        err.setText("");
    } catch(NumberFormatException e){
            alab.setText("");
            blab.setText("");
            anslab.setText("");
            err.setText("Enter only Integers!");
        } catch(ArithmeticException a){
            alab.setText("");
            blab.setText("");
            anslab.setText("");
            err.setText("");
        }
    }
```

Teacher's Signature :

```
jform.setVisible(true);
}

public static void main(String args[]){

    SwingUtilities.invokeLater(new Runnable() {
        public void run() {

            new SwingDemo();
        }
    });
}
```

Expt. No. ---------- Page No. ----------

Lab:- 10 programs: (Inter process communication)

```
class Q {
private int n;
private boolean valueset = false;
synchronized int get(){
   while(!valueSet){
      try {
         System.out.println("\n consumer waiting");
         wait();
         } catch (Interupted_Exception e){
            system.out.println("InterruptedException
                                caught");

      }
   }
   system.out.println("Got:"+n);
   valueSet= false;
   System.out.println("Notifying producer");
   notify();
   return n;
   }

Synchronized void put(int n){
   while(ValueSet){
      try{
         System.out.println("\nProducer waiting");
         wait()
      }
```

Teacher's Signature : _____

```
catch (InterruptionException e){
    while (valueSet){
        try{
            System.out.println("\n Producer waiting");
                                "Interrupted Exception");
                                }
    }
    this.n = n
    valueSet = true;
    System.out.println("Put :" + n);
    System.out.println("Notifying consumer");
    notify();
  }
}

class pro consumer implements Runnable{
    private Q q;
    consumer (Q q){
        this.q = q
        new Thread( this, "consumer").start();
    }

public void run(){
    int i=0;
    while( i<15){
        int r = q.get();
        System.out.println(" consumed:" +r);
        i++
    }
}
```

Expt. No. ---------- Page No. ----------

```
Public class PCFixed{
  public static void main (String args[]){
     Q q = new Q();
    new Producer(q);
    new Consumer(q);
    System.out.println("press control-c to stop.");
    }
```

output:

Put:0
Notifying consumer
producer waiting
Got:0.
Notifying producer
Put 1
Notifying consumer
producer waiting
Consumed: 1
Put 2
Notifying consumer
Got: 2
Notifying producer
consumed: 2

6/2/24

Expt. No. ..... Date .....

Page No. .....

Lab 10 deadlock (continuation)

```
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();
        }
}

classB {
    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(name)
            a.last()
        }
```

Teacher's Signature : _____

```
synchronized void last(){
    System.out.println("Inside B.last");
}}

class Deadlock implements Runnable {
    A a = new A();
    B b = new B();

    Deadlock(){
        Thread.currentThread().setName("MainThread");
        Thread t = new Thread();
        t.start();
        a.foo(b);
        System.out.println("Back in main thread");
    }

public void run(){
    b.bar(a);
    System.out.println("Back in other thread");
}

public static void main(String args[]){
    new Deadlock();
}}
```

Date

Page No.

Sub. No.

Output

Racing Thread entered B.bar
Main Thread entered A.foo
Main Thread trying to call B.last(),
Racing thread trying to call A.last();

13/2/24

Teacher's Signature :

LAB: 1

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

```
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>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<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: 2

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

```
import java.util.Scanner;
class Student{
   String name;
   String usn;
   double SGPA;
   Subject subject[];
   Scanner s;

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

   }

   void getStudentDetails(){
      System.out.println("Enter your Name:");
      name=s.next();
      System.out.println("Enter your USN:");
      usn=s.next();
   }

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

         subject[i].grade=(subject[i].subjectMarks/10)+1;

         if (subject[i].grade==11){
            subject[i].grade=10;
         }
         if (subject[i].grade<=4){
            subject[i].grade=0;
         }

      }
   }

   void computeSGPA(){
      int effectiveScore=0;
      int totalCreadits=0;
```

```
      for(int i=0;i<9;i++){
         effectiveScore += (subject[i].grade*subject[i].credits);
         totalCreadits += subject[i].credits;
      }

      SGPA=(double)effectiveScore/(double)totalCreadits;
   }
}
class Subject{
          int subjectMarks;
          int credits;
          int grade;

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

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

LAB: 3

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

```
import java.util.Scanner;
class Book{
   String name, author;
   int price,page_num;

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

   String toStrings(){
      String name,author,price,page_num;
      name="Book name: "+this.name+"\n";
      author="Author name: "+this.author+"\n";
      price="Price: "+this.price+"\n";
      page_num="Number of pages: "+this.page_num+"\n";

      return (name+author+price+page_num);
   }

   public static void main(String args[]){
      int n;
      String name,author;
      int price,page_num;

      Scanner sc=new Scanner(System.in);

      System.out.println("Enter the no. of books:");
      n=sc.nextInt();

      Book b[]=new Book[n];

      for (int i=0;i<n;i++){
         System.out.println("Enter name of book:");
         sc.nextLine();
         name=sc.nextLine();

         System.out.println("Enter author of a book:");
         author=sc.nextLine();

         System.out.println("Enter the price of book:");
         price=sc.nextInt();

         System.out.println("Enter the no. of pages of book:");
         page_num=sc.nextInt();
```

```
            b[i]=new Book(name,author,price,page_num);
        }

        System.out.println("Book Details:");
        for(int i=0;i<n;i++){
            System.out.println("Book "+(i+1)+" :\n"+b[i].toStrings());
        }
    }
}
```

LAB: 4

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

```
import java.util.Scanner;
public class InputScanner {
	Scanner s;
	InputScanner(){
		s=new Scanner(System.in);
	}
}
abstract public class Shape extends InputScanner{
	double a,b;
	abstract void getInput();
	abstract void displayArea();
}
public class Rectangle extends Shape {
	void getInput() {
		System.out.println("Enter the dimension of rectangle:");
		a=s.nextDouble();
		b=s.nextDouble();
	}

	void displayArea() {
		System.out.println("Area of reactangle:"+(a*b));
	}
}
public class Circle extends Shape{
	void getInput() {
		System.out.println("Enter the dimension of circle:");
		a=s.nextDouble();
	}

	void displayArea() {
		System.out.println("Area of circle:"+(3.14*a*a));
	}
}
public class Triangle extends Shape {
	void getInput() {
		System.out.println("Enter the dimension of triangle:");
		a=s.nextDouble();
		b=s.nextDouble();
	}

	void displayArea() {
		System.out.println("Area of triangle:"+((a*b)/2));
	}
}
public class MainMethod {

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

```
                Rectangle R=new Rectangle();
                R.getInput();
                R.displayArea();

                Triangle T=new Triangle();
                T.getInput();
                T.displayArea();

                Circle C=new Circle();
                C.getInput();
                C.displayArea();
        }

}
```

LAB: 5

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

```
import java.util.Scanner;
public class Account {
	String customer_name;
	int account_no;
	String type_acc;
	double balance=0;
	Scanner sc;

	Account(String customer_name,int account_no,String type_acc){
		this.customer_name=customer_name;
		this.account_no=account_no;
		this.type_acc=type_acc;
		sc=new Scanner(System.in);
	}

	void diposite() {
		System.out.println("Enter the deposit amount:");
		int dipo=sc.nextInt();
		balance+=dipo;
	}

	void withdrawal() {
		System.out.println("Enter the withdrawl amount:");
		int with=sc.nextInt();
		if(with>balance) {
			System.out.println("Insufficient Balance");
		}
		else{
			balance-=with;
		}
	}

	void display() {
		System.out.println("Customer name:"+customer_name);
		System.out.println("Account number:"+account_no);
		System.out.println("Type of Account:"+type_acc);
		System.out.println("Balance:"+balance);
	}
	void applyinterest() {}

}

public class Cur_acct extends Account {

	Cur_acct(String customer_name,int account_no,String type_acc){
		super(customer_name,account_no,type_acc);}
```

```
	void withdrawal() {
		System.out.println("Enter the withdrawl amount:");
		int with=sc.nextInt();
		if (balance<=2000) {
			double pen=balance/(0.06);
			System.out.println("Insuffiecient balance penalty to be paid:"+pen);
			balance+=pen;
		}
		else{
			balance-=with;
		}
	}
}

public class Sav_acct extends Account{
Sav_acct(String customer_name,int account_no,String type_acc){
super(customer_name,account_no,type_acc);
}

void applyinterest() {
System.out.println("Enter the interest rate:");
int rate=sc.nextInt();
double interest=balance*rate;
balance+=interest;
System.out.println("Balance after interest:"+balance);
}
}

import java.util.Scanner;
public class Bank {
	public static void main(String[] args) {
		Scanner sc=new Scanner(System.in);
		System.out.println("Enter customer name:");
		String cus_name=sc.nextLine();
		System.out.println("Enter account number:");
		int acc_no=sc.nextInt();

		Account ca=new Cur_acct(cus_name,acc_no,"Current Account");
		Account sa=new Sav_acct(cus_name,acc_no,"Saving Account");

		int choice;
		while(true){
			System.out.println("----MENU----");
			System.out.println("1.Deposite\n2.Withdrawl\n3.Compute intereset for Saving
			account\n4.Display account details\n5.Exit");
			choice=sc.nextInt();

			switch(choice) {
			case 1:
			System.out.println("Enter the type of account:\n1.Saving Account \n2.Current Account");
			int acc=sc.nextInt();
			if(acc==1) {
```

```
                    sa.diposite();
                    }
                    else {
                    ca.diposite();
                    }
                    break;
                    case 2:
                    System.out.println("Enter the type of account:\n1.Saving Account \n2.Current Account");
                    int acc1=sc.nextInt();
                    if(acc1==1) {
                    sa.withdrawal();
                    }
                    else {
                    ca.withdrawal();
                    }
                    break;
                    case 3:
                    sa.applyinterest();
                    break;
                    case 4:
                    System.out.println("Enter the type of account:\n1.Saving Account \n2.Current Account");
                    int acc2=sc.nextInt();
                    if(acc2==1) {
                    sa.display();
                    }
                    else {
                    ca.display();
                    }
                    break;
                    case 5:
                    break;
                    default:
                    System.out.println("Invalid Choice");
                    break;
                    }

                    if(choice==5) {
                    break;
                    }
                }
        }
}
```

LAB: 6

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

```
package CIE;
import java.util.Scanner;
public class Student{
   public String usn,name;
   public int sem;
   public void inputStudentDetails(){
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter the student usn:");
      usn=sc.nextLine();
      System.out.println("Enter the student name:");
      name=sc.nextLine();
      System.out.println("Enter student semester:");
      sem=sc.nextInt();
   }

   public void disply(){
      System.out.println("Student USN:"+usn);
      System.out.println("Student Name:"+name);
      System.out.println("Student Sem:"+sem);
   }
}

package CIE;
import java.util.Scanner;
public class Internals extends Student{
   public int marks[]=new int[5];
   public void inputCIEmarks(){
      Scanner sc=new Scanner(System.in);
      for(int i=0;i<5;i++){
         System.out.println("Enter the marks for subject "+(i+1)+":");
         marks[i]=sc.nextInt();
      }
   }
}
package SEE;
import CIE.Internals;
import java.util.Scanner;
public class Externals extends CIE.Internals{
   public int marks[];
   public int finalMarks[];

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

```
    public void inputSEEMarks(){
        Scanner sc=new Scanner(System.in);
        for(int i=0;i<5;i++){
            System.out.println("Enter subject "+(i+1)+" marks:");
            marks[i]=sc.nextInt();
        }
    }

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

    public void displayFinalMarks(){
        inputStudentDetails();
        for(int i=0;i<5;i++){
            System.out.println("Subject "+(i+1)+" final marks:"+finalMarks[i]);
        }
    }
}
import SEE.Externals;
class PackageMain{
    public static void main(String args[]){
        int numOfStudent=2;
        Externals finalMarks[]=new Externals[numOfStudent];
        for(int i=0;i<numOfStudent;i++){
            finalMarks[i]=new Externals();
            finalMarks[i].inputStudentDetails();
            System.out.println("Enter CIE Marks:");
            finalMarks[i].inputCIEmarks();
            System.out.println("Enter SEE Marks:");
            finalMarks[i].inputSEEMarks();
        }
        System.out.println("Displaying data:\n");
        for(int i=0;i<numOfStudent;i++){
            finalMarks[i].calculateFinalMarks();
            finalMarks[i].disply();
            finalMarks[i].displayFinalMarks();
        }
    }
}
```

LAB: 7

Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called "Father" and derived class called "Son" which extends 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.

```
public class WrongAge extends Exception{
   WrongAge(String msg){
      super(msg);
   }
}
import java.util.Scanner;
class InputScanner{
   Scanner sc;
   InputScanner(){
      sc=new Scanner(System.in);
   }
}
class Father extends InputScanner{
   int fatherAge;
   Father() throws WrongAge{
      System.out.println("Enter father's age:");
      fatherAge=sc.nextInt();
         if(fatherAge<0){
         throw new WrongAge("Age cannot be negative");
      }

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

   }

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

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

```
        try{
            Son son=new Son();
            son.display();
        }
        catch(WrongAge e){
            System.out.println(e);

        }
    }
}
```

LAB: 8

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 BMS_College_of_Engineering implements Runnable {
    public void run() {
        while (true) {
            try {
                System.out.println("BMS College of Engineering");
                Thread.sleep(10000); // Sleep for 10000 milliseconds (10 seconds)
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}

class CSE implements Runnable {
    public void run() {
        while (true) {
            try {
                System.out.println("CSE");
                Thread.sleep(2000); // Sleep for 2000 milliseconds (2 seconds)
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}

public class ThreadMain {
    public static void main(String[] args) {
        Thread t1 = new Thread(new BMS_College_of_Engineering());
        Thread t2 = new Thread(new CSE());
        t1.start();
        t2.start();
    }
}
```

LAB: 9

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.

```
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!");
        }
      }
    });

    // 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();
      }
    });
  }
}
```

LAB: 10
Demonstrate Inter process Communication and deadlock.

```
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<5) {
			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<5) {
			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:

```
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); // get lock on a in this thread.
            System.out.println("Back in main thread");

        }
        public void run() {
            b.bar(a); // get lock on b in other thread.
            System.out.println("Back in other thread");
        }

        public static void main(String args[]) {
            new Deadlock();
        }

}
```