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:

Sanghamitra R
1BM22CS237

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

INDEX

NAME : Sanghamitra R ~~CLS~~ : UOJ SEC : _____ ROLL NO : _____

S. No.	Date	Title	Page No.	Teacher's Sign / Remarks
01	12/12/23	Quadratic Eq		12-12
02	19/12/23	Classes		19-12
03	26/12/23	Book - information		26-12
04	02/01/24	Area of Rectangle, Triangle, Circle		2-1
05	16/01/24	Bank Account Class		16-1
06	23/01/24	Strings and Generics		23-1
07	23/01/24	CIE and SEE package		23-1
08	30/01/24	Wrong Age		30-1
09	06/02/24	Threads		6-2
10	13/02/24	IPC and deadlock		
11	20/02/24	Division of integers (VI)		20/2/2024

Date
Page

LAB 1 Sanghamitra R - 1BM22CS237

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.

```
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 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("Root 1= Root 2 =" +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 + "Root 2 = " + r2);
}
else if(d<0)
{
  System.out.println("Roots are imaginary");
  r1 = (-b)/(2*a);
  r2 = Math.sqrt(-d)/(2*a);
  System.out.println("Root 1 =" +r1+"i" +r2);
  System.out.println("Root 1=" +r1+"-i" +r2);
}
}
}
class QuadraticMain
{
  public static void main(String args[])
  {
    Quadratic q = new Quadratic();
    q.getd();
    q.compute();
  }
}
```

OUTPUT:

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

Enter coefficients of a, b, c:
1
5.
6
Roots are real and distinct
Root 1 = -2.0 Root 2 = -3.0

Enter coefficients of a, b, c:
3
4
5
Roots are imaginary
Root 1 = 0.0 + i 1.1055415967851332
Root 2 = 0.0 - i 1.1055415967851332

Sanjhamitra R - 1BM22CS237

12/12/23

LAB 02 Q) Develop a Java program to create a class Student with members USN, name, an array credits & an array marks. Include methods to accept & display details & a method to calculate SGPA of a student.

```
import java.util.Scanner;
class Subject {
{
  int subMarks;
  int credits;
  int grade;
}

~~Studs~~ class Student
{
  Subject subject[];
  String name;
  String usn;
  double SGPA;
  Scanner s;
  int i;

  Student()
  { int i;
    subject = new Subject [8];
    for(i=0; i<8; i++)
    subject [i] = new Subject();
    s = new Scanner (System.in);
  }
  void getStudentDetails()
  {
  System.out.println("Enter your name: ");
  name = s.next();
  System.out.println("Enter your usn: ");
  usn = s.next();
  }
  void getMarks()
  {
  for( i=0; i<8; i++)
  {
  System.out.println ("Enter marks of subject"+(i+1)+" :");
```

```
subject[i].subMarks = S.nextInt();
System.out.println("Enter credits of Subject"+(i+1)+":");
subject[i].credits = S.nextInt();
subject[i].grade = (subject[i].subMarks/10)+1;
}
}
Void computeSGPA()
{
  int effscore=0;
  int totalcreds=0;
for(i=0;i<8;i++)
{
  effscore+=subject[i].grade*subject[i].credits;
  totalcreds+=subject[i].credits;
}
SGPA=(double)effscore/(double)totalcreds;
}
}
Class SGPA
{
public static void main(String args[])
{
 student S1 = new student();
 S1.getMarks();
 S1.computeSGPA();
 System.out.println("Name:"+S1.name);
 System.out.println("USN:"+S1.usn);
 System.out.println("SGPA:"+S1.SGPA);
}
}
```

OUTPUT:

```
Enter your name: Sanga
Enter your usn: 237

Enter marks of subject 1:
90
Enter credits of subject 1:
4
Enter marks of subject 2:
90
Enter credits of subject 2:
4
Enter marks of subject 3:
85
Enter credits of subject 3:
3
Enter marks of subject 4:
85
Enter credits of subject 4:
2
Enter marks of subject 5:
79
Enter credits of subject 5:
3
Enter marks of subject 6:
78
Enter credits of subject 6:
1
Enter marks of subject 7:
90
Enter credits of subject 7:
1
Enter marks of subject 8:
92
Enter credits of subject 8:
2

Name: sanga
USN: 237
SGPA: 9.35
```

Done
14/12 -Wr

Date 26/12/23
Page

LAB-03: 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.*;

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

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

public void set(String name, String author, int price,
        int Pages) {
    this.name = name;
    this.author = author;
    this.price = price;
    this.numPages = numPages;
}
```

```
public String toString() {
    String name = "Book name: " + this.name + "\n";
    String author = "Author name: " + this.author + "\n";
    String price = "price: " + this.price + "\n";
    String numPages = "Number of pages: " + this.numPages
                      + "\n";
    return name + author + price + numPages;
}

public String getName() {
    return this.name;
}
public String getAuthor() {
    return this.author;
}

public int getPrice() {
    return this.price;
}
public int getNumberOfPages() {
    return this.numPages;
}

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

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

```
for(int i=0 ; i<n ; i++) {
    System.out.print("Enter" + (i+1) + "Book name,
        author name, price. and no.of pages:");
    String name = sc.next();
    String author = sc.next();
    int price = sc.nextInt();
    int numofpages = sc.nextInt();

    b[i] = new Book(name, author, price, numPages);
}
    for(int i=0; i<n; i++) {
        System.out.println(b[i].toString());
    }
}
```

OUTPUT:

Enter no. of books : 2
Enter 1 book name, author name, price & number of pages : book 1 Sanga 1400 68

Enter 2 book name, author name, price & no. of pages :
book 2 Sanjana 1600 70

Book Name : Book 2
Author Name : Sanga
Price : 1600
Number of pages : 700

Book Name : book 2
Author name : Sanjana
Price : 870
Number of pages : 200

26/02/23
Sangramitra R
1BM22CS237

02/01/24

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, Circle such that each one of the classes extends class shape. Each one of the classes contains only the method printArea() that prints area of given shape.

```
import java.util.Scanner;

class InputScanner{
    int d1, d2;
    Scanner s = new Scanner (System.in);
    InputScanner(){
        if (this.getClass() == circle.class)
        {
            System.out.println ("Enter radius of circle\n");
            d1 = s.nextInt();
        }
        else
        {
            System.out.println("Enter height & width\n");
            d1 = s.nextInt();
            d2 = s.nextInt();
        }
    }
}
abstract class Shape extends InputScanner()
{
    abstract void printArea();
}
```

Date
Page

```
class triangle extends Shape {
    void PrintArea() {
        System.out.println("Area of triangle is" + (double) (d1*d2)/2);
    }
}

class rectangle extends Shape {
    void PrintArea() {
        System.out.println("Area of rectangle is:" + (double) (d1*d2));
    }
}

class circle extends Shape {
    void PrintArea() {
        System.out.println("Area of circle is:" + (double) (3/4 * d1*d2));
    }
}

Class Area Main {
    public static void main (String args[]) {
        Rectangle r = new Rectangle();
        Triangle t = new Triangle();
        Circle c = new Circle();
        s.PrintArea()
        t.PrintArea()
        c.PrintArea()
    }
}
```

OUTPUT:

Enter height and width:
2
4
Enter height and ~~weight~~ width:
6
7
Enter radius of circle:
5
Area of rectangle is : 8.0
Area of Triangle is : 21.0
Area of circle is : 78.5
Sanghamitra R
1BM22CS237

16-1-24

Date
Page

LAB 05

Develop a java program to create a class to account that stores customer name, account number, and type of account. From this derive the classes current acc and savings account to make them more specific to their requirements.
Achieve the following tasks:
(a) Accept deposit from customer & update the balance
(b) Display the balance
(c) Compute & deposit interest
(d) Permit withdrawal and update the balance.

```
import java.util.Scanner;

class Input {
    Scanner sc = new Scanner(System.in);
}

class Account extends Input {
    String name;
    int accno.;
    double balance;

    void getDetails(){
        System.out.println("Enter name: ");
        name = sc.nextLine();
        System.out.println("Enter acc no.:");
        accNo. = sc.nextInt();
    }
```

```
void deposit(){
    System.out.println("Enter amnt to deposit:");
    double amnt = sc.nextDouble();
    balance += amnt;
    System.out.println("Amount Deposited:");
}

void withdrawn(){
    System.out.println("Enter amount");
    double amnt = sc.nextDouble();
    if(balance >= amnt){
        balance-= amnt;
        System.out.println("Amount withdrawn");
    }
    else{
        System.out.println("Insufficient balance");
    }
}

void Display(){
    System.out.println("Name:" + name);
    System.out.println("Account Number:" +
                                    accNO);
    System.out.println("Balance:" + balance);
}
}

Class Current extends Account{
    double minbal = 500;
    double penalty = 100;
```

```
void withdraw(){
    super.withdraw();
    checkMinBalance();
  }

  private void checkMinBalance(){
      if(balance < minBal){
          balance -= Penalty;
          System.out.println("Penalty applied
              for low balance");
      }
  }
}

class Savings extends Account{
    double interestRate = 0.04;

  void computeInterest(){
    double interest = balance * interestRate;
    balance += interest;
    System.out.println("Interest credited :" + interest);
  }
}

class Bank extends Account{
  public static void main(String[] args)
  {
    savings ob1 = new Savings();
    current ob2 = new Current();

    ob1.getDetails();
```

```
ob2.getDetails();
int choice;
String acc;
System.out.println("Menu:");
System.out.println("1. Deposit \n 2. Withdraw \n
3. Display Balance \n 4. Compute Interest (Saving
Only) \n 5. Exit \n");

do{
  System.out.print("Enter your choice:");
  choice = sc.nextInt();
  System.out.print("Enter the acc type");
  acc = sc.next();
  switch(choice){
    case 1:
      if(acc.equals("Saving"))
        ob1.deposit();
      else
        ob2.deposit();
      break;
    Case 2:
      if(acc.equals("Savings"))
        ob1.withdraw();
      else
        ob2.withdraw();
      break;

    Case 3:
      if(acc.equals("Saving"))
        ob1.display();
      else
        ob2.display();
      break;
```

```
case 4:
    ob1.computeInterest();
    break;

case 5:
    break;
    Default:
        System.out.println("Invalid choice");
    }
  }while(choice != 5);
 }
}
```

OUTPUT:

Enter name: Sanghamitra
Enter accNo: 1
Enter name: ~~Shoo~~ Samrat
Enter accNo: 2

Menu
1. Deposit
2. Withdraw
3. Display
4. Compute Interest (savings only)
5. Exit

Enter your choice: 1
Enter acc type: Savings
Enter deposit amount: 1000
Enter your choice: 1
Enter acc type: current
Enter deposit amount: 5000

Enter your choice : 2
Enter acc type : savings
Enter withdraw amt : 500
Enter your choice : 2
Enter acc type : current
Enter withdraw amt : 4600
Penalty applied
Enter your choice : 3
Enter acc type : current
Name : Sheeyas
Acc no. : 2
Balance : 300

Enter your choice : 4
Interest credited : 40
Enter your choice : 2
Enter acc type : savings
Enter withdraw amt : 1050
insufficient balance

Sanghamitra R
1BM22CS237

16.1.24

Date
Page

LAB 6: Strings & Generics

1. OUTPUT: Hello, World // string constructors

2. Hello World, BMSCE // length, literal, concat
5

3. 20 // to String()
101 Samraat

4,5) //use getClass() & getBytes()
BMSCE
65
66
67
68
69
70
71
ABCDEFG //toCharArray()

6,7) equals and region matches
BMSCE equals BMSCE → True
BMSCE equals College → false
BMSCE equals BMSCE → false
BMSCE equalsIgnoreCase Bmsce → True

8,9,10) startsWith(), endsWith() & equals() vs ==
True
True
false
true

Hello equals Hello → true
Hello == Hello

11) Sorting using compare to ()
Apple
Bones
Corn
Dog
Eee
Zebra

String Buffer Functions

O/P Output:

buffer before = Hello
charAt(1) = e
buffer after = Hi
charAt(1) after = i

f //getChar()

Hello BMS

a = 30!

I like Java!

!ava J ekil I

After delete : This is test
After delete charAt : his a test
After replace : his was a test

2. Generics

Enter elements is integer array
1 2 3 4 5

Enter elements in second stack
10.0 20.0 30.0 40.0 50.0

Elements of stack 1	Elements of Stack 2
5	50.0
4	40.0
3	30.0
2	20.0
1	10.0

Sanghamitra R
1BM22CS237

30-1-24

LAB 6 → CIE and SEE package

```
//Student.java

package CIE;
import java.util.Scanner;

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

void inputStudentDetails(){
    Scanner S=new Scanner(System.in)
    usn = S.next();
    iname = S.next();
    sem = S.nextInt();
}
Student(){
}
Student(String usn, String name, int sem){
    this.usn = usn;
    this.name = name;
    this.sem = sem;}}
```

~~Internal~~

Date
Page

// internals Java

```
import java.util.Scanner;
package CIE;

public class Internals extends Student {
  protected int[] marks = new int[5];
  String name;
  String usn;
  int sem;

void inputCIEdetails(){

   Scanner S= new Scanner(System.in);
   System.out.println("Enter internals marks in
subjects");

for(int i=0; i<5; i++){
     System.out.println("Sub " + (i+1));
    marks[i] = S.nextInt();
 }
}
Internals() {
}
Internals(String usn, String name, int sem, int marks[]){

 Super(usn, name, usn);

for(int i=0; i<5; i++){
   this.marks[i]= marks[i]; }
```

```
void displayStudentDetails(){
    System.out.println("Name" + this.name + "\n"
    + this.usn + "sem:" + this.sem);
    System.out.println("marks:");
    for(int i=0; i<5; i++){
        System.out.println("subject " + (i+1) + ":"
        marks[i]);
    }
}
}

//Externals.java

package SEE;
import CIE.internals;

import java.util.Scanner;

public class External extends Internals{

    protected int[] marks1 = new int[5];
    protected final Marks = new int[5];

External(){
}
public void inputSEEmarks(){
    System.out.println("Enter SEE marks");
for(int i=0; i<5; i++)
{
    System.out.println("subject" + (i+1) + "marks");
    marks2[i] = s.nextInt();
}
```

```
public void calcFM() {
    for(int i=0; i<5; i++){
      final Marks = marks1[i]/2 + Super.marks[i];
    }
}

public void displayFinalMarks() {
    displayStudentDetails();

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

//

main.java

```
import SEE.externals;
import java.util.Scanner;

class main {
  public static void main(String args[]){
    Scanner s = new Scanner(System.in);
    int n;
    System.out.println("Enter no. of students");
    n = s.nextInt();
    External finalMarks[] = new External[n];
```

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

    finalMarks[i] = new External();
    finalMarks[i].inputStudentDetails();
    finalMarks[i].inputCIEDetails();
    finalMarks[i].inputSEEMarks();
}

System.out.println("final details:");
for(i=0; i<n; i++){
    finalMarks[i].calcFinalMarks();
    finalMarks[i].displayFinalMarks();
}

}
```

OUTPUT:

```
No. of students : 1
Enter USN : 237
Enter name : Sanghamitra

Semester : 3

Enter CIE marks:
Subject 1 : 22
Subject 2 : 27
Subject 3 : 34
Subject 4 : 37
Subject 5 : 36
```

Date
Page

Enter SEE Marks:

Subject 1: 90
Subject 2: 93
Subject 3: 91
Subject 4: 94
Subject 5: 93

Displaying data:

Name: Sanghamitra K
USN: 237
Subject 1: 78
Subject 2: 72
Subject 3: 83
Subject 4: 84
Subject 5: 83
Sanghamitra K
1BM22CS237

10/1/24

LAB 7: WAP to demonstrate the exception handling in inheritance tree.

```
import java.util.*;

class WrongAge extends Exception{
    WrongAge(String s){
        super(s);
    }
}

class InputScanner{
    Scanner s = new Scanner(System.in);
}

class father extends InputScanner{
    int fatherAge;

father() throws WrongAge{
    Sys.out.println("Enter father age:");
    fathersAge = S.nextInt();

    if(fathersAge<0)
        throw new WrongAge("Age cannot be
            negative");
}
void display(){
    System.out.println("Father's age:"+fatherAge);
}
}
```

Date
Page

```
class Son extends Father {
    int SonAge;

Son() throws WrongAge {
    super();
  System.out.println("Enter son's age:");
  SonAge = s.nextInt();

  if (sonAge < 0)
    throw new WrongAge("Age cannot be -ve");

else-if (sonAge > FatherAge)
  throw new WrongAge("Son's age cannot be
    greater than father's age");

else {
    displayF();
    displaySon();

  throw new WrongAge("Valid Age");
  }
}

void displaySon() {
    System.out.println("Son's age:" + SonAge);
  }
}
```

```
class exceptions main{
    public static void main (String [] args)
{
try {
    Son son = new Son();
  }
  catch(WrongAge e){
    System.err.println (e.getMessage());
  }
}
}
```

OUTPUT:

Enter father's age:
44
Enter Son's age:
23

Father's age: 43
Son's age: 23
Valid Age

Enter father's age:
-2
Age cannot be negative

Enter father's age:
44
Enter Son's age
66
Son's age cannot be greater than father's age.

Sanghamitra K
1BM22CS237

Date
Page

LAB8: WAP which creates 2 threads, one thread displaying "BMS College of Engineering" once every 10 seconds & another displaying "CSE" once every 2 seconds

```
import java.lang.*;

Class DisplayMessageThread extends Thread{
    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(InterruptException e){
            System.out.println(Thread.currentThread().
                getName() + "interrupted");
        }
    }
}

public class Main{
    public static void Main(String[] args){
        DisplayMessageThread t1 = new DisplayMessage
                Thread("BMS College of Engg", 10000);
```

```
DisplayMessageThread t2 = new DisplayMessageThread
                          ("CSE", 2000);
t1.setName("First Thread");
t2.setName("Second Thread");
t1.start();
t2.start();

Try{
  Thread.sleep(30000);
}
catch (InterruptException e){
  System.out.println("Main thread interrupted");
}
t1.interrupt();
t2.interrupt();

System.out.println("Main Thread exiting!");
}
}
```

OUTPUT:

BMS College Of Engineering
CSE
CSE
CSE
CSE
CSE
BMS CE
CSE
CSE
CSE
CSE
CSE
BMSCE

CSE
CSE
CSE
CSE
CSE
Main thread exiting
Second thread interrupted
First thread interrupted
Sanghamitra R
1BM22CS237

LAB-09 Demonstrate inter process Communicatio & deadlock

```
class Q {
    int n;
    boolean valueset = false;

    while(!valueset) {
      try {
        System.out.println("Consumer waiting");
        wait();
      }
      catch (InterruptException e) {
        System.out.println("InterruptException
          caught");
      }
    }
    System.out.println("Got:" + n);
    valueset = false;
    System.out.println("Intimate producer");
    notify();
    return n;
  }
  Synchronised void put(int n) {
    while(valueset) {
      try {
        System.out.println("producer waiting");
        wait();
      }
      catch (InterruptException e) {
        System.out.println("Interrupt Exception
          caught");
      }
    }
```

```
        this.n = n;
        valueset = true;
        System.out.println("Put: " + n);
        System.out.println("Intimate Consumer");
        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();
            i++;
        }
    }
}
```

PCFixed

```
class PCFixed{
  public static void main(String[] args){
    Q q = new Q();
    newProducer(q);
    newConsumer(q);
    System.out.println("Press Control to stop");
  }
}
```

OUTPUT:

Put : 0
Got : 0
Put : 1
Got : 1
Put : 2
Got : 2

Sanghamitra R IBM22CS237

20/2/2024

Date
Page

Deadlock

```
class A {
    synchronised 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 {
    Synchronised 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("Main Thread");
        Thread t = new Thread(this, "Racing Thread");
        t.start();
        a.foo(b);
        System.out.println("Back is main thread");
    }

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

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

Date
Page

OUTPUT:

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

20/2/2024 Sanghamitra R
IBM22CS237

LAB10: Write a program that creates a use
interface to perform integer divisions
The user enters two nos. in the text fie
Num1 and Num2. The division of Num1 & N
is displayed in the Result field when the div
button is clicked.
If Num 1 and Num 2 were not an integ
the prog would throw a NumberFormatExcep
If Num2 were zero, the program would th
an Arithmatic Exception Display the exception
message dialog box.

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

public class DivisionMain1 extends Frame imple
                                    ActionListe
{
   TextField num1, num2;
   Button dResult;
   Label outResult;
   String out = " ";
   double resultNum;
   int flag = 0;

public DivisionMain1()
{
   setLayout(new FlowLayout());
   dResult = new Button("RESULT");
   Label number1 = new Label("Number 1:", Label
   Label number2 = new Label("Num 2:", Label.RIGHT);
   num1 = new TextField(5);
   num2 = new TextField(5);
```

```
outResult = new Label("Result:", Label.RIGHT);

add(number1);
add(num1);
add(number2);
add(num2);
add(dResult);
add(outResult);

num1.addActionListener(this);
num2.addActionListener(this);
dResult.addActionListener(this);

addWindowListener(new WindowAdapter())
{
  public void windowClosing(WindowEvent we)
  {
    System.exit(0);
  }
});
}
public void actionPerformed(ActionEvent ae)
{
  int n1, n2;
  try
  {
    if(ae.getSource()==dResult)
    {
      n1=Integer.parseInt(num1.getText());
      n2=Integer.parseInt(num2.getText());
/*
```

```
if(n2==0)
    throw new ArithmeticException();
  out = n1+" "+n2;
  resultNum = n1/n2;
  out += String.valueOf(resultNum);
  repaint();
 }
}

catch(NumberFormatException e1)
{
  flag = 1;
  out = "Number Format Exception!"+e1;
  repaint();
}

catch(ArithmeticException e2)
{
  flag = 1;
  out = "Divide by 0 Exception!"+e2;
  repaint();
 }
}

public static void main(String[] args) {

  DivisionMain1 dm = new DivisionMain1();
  dm.setSize(new Dimension(800,400));
  dm.setTitle("Division Of Integers");
  dm.setVisible(true);
 }
}
```

Date
Page

OUTPUT:

Number 1: 21

```
public void paint(Graphics g)
{
  if(flag==0)
  g.drawString(out, outResult.getX() + outResult.getWidth()
    outResult.getY() + outResult.getHeight() - 8);

  else
  g.drawString(out, 100, 200);
  flag=0;
}
```

OUTPUT:

Number 1: [12] Number 2: [6] Res: 12 6 2.0

Sanghamitra R

1RN22CS237

Functions used:

A button is a control component with a label that generates an event when pushed.

SetLayout() method allows you to set the layout of the container

addActionListener() is a type of class in Java that receives a notification whenever any action is performed.

repaint() method is an asynchronous method applet class.

setSize() sets the size of this Dimension object to the specified width and height

setTitle() function defines the title to appear at the top of the sketch window.

setVisible() method makes the frame appear on the screen.

20/02/2024

1)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

```
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("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 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);
   System.out.println("Root1="+r1+"-i"+r2);
  }
 }
}
class QuadraticMain
{
 public static void main(String[] args)
 {
  Quadratic q=new Quadratic();
  q.getd();
  q.compute();
 }
}
```

2)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.

```
import java.util.Scanner;

class Subject
{
	int subMarks;
	int credits;
	int grade;
}

class Student
{
	Subject subject[];
	String name;
	String usn;
	double SGPA;
	Scanner s;
	int i;

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

```
	}

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

	void getMarks()
	{
		for(i=0;i<8;i++)
		{
			System.out.println("Enter marks of subject " + (i+1) + ": ");
			subject[i].subMarks = s.nextInt();
			System.out.println("Enter credits of subject " + (i+1) + ": ");
			subject[i].credits = s.nextInt();
			subject[i].grade = (subject[i].subMarks/10) + 1;
		}
	}
	void computeSGPA()
	{
		int effscore = 0;
		int totalcreds = 0;
		for(i=0;i<8;i++)
		{
			effscore += subject[i].grade * subject[i].credits;
			totalcreds += subject[i].credits;
		}
		SGPA = (double)effscore/(double)totalcreds;
	}
}

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);
        }
}
```

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 Books
{
        String name;
        String author;
        int price;
        int numPages;

        Books(String name, String author, int price, int numPages)
        {
                this.name = name;
                this.author = author;
                this.price = price;
                this.numPages = numPages;
        }

        public String toString()
        {
                String name, author, price, numPages;
                name = "Book name: " + this.name + "\n";
                author = "Author name: " + this.author + "\n";
                price = "Price: " + this.price + "\n";
                numPages = "Number of Pages: " + this.numPages + "\n";
                return name + author + price + numPages;
        }
}

class books_main
{
        public static void main(String args[])
        {
                Scanner s = new Scanner(System.in);
                int n,i;
```

```
                String name;
                String author;
                int price;
                int numPages;

                System.out.println("Enter number of books: ");
                n = s.nextInt();

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

                for(i = 0; i < n; i++)
                {
                        System.out.println("Enter name of book: ");
                        name = s.next();
                        System.out.println("Enter author of book: ");
                        author = s.next();
                        System.out.println("Enter price of book: ");
                        price = s.nextInt();
                        System.out.println("Enter number of pages: ");
                        numPages = s.nextInt();
                        b[i] = new Books(name,author,price,numPages);
                }

                for(i = 0; i < n; i++)
                {
                        System.out.println(b[i].toString());
                }

        }
}
```

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 ofthe given shape.

```
import java.util.Scanner;

class InputScanner{
int d1, d2;
Scanner sc = new Scanner(System.in);
InputScanner(){
```

```
if(this.getClass() == Circle.class){
System.out.println("Enter radius of circle: ");
d1 = sc.nextInt();
}
else{
System.out.println("Enter height and weight: ");
d1 = sc.nextInt();
d2 = sc.nextInt();
}
}
}

abstract class Shape extends InputScanner{
abstract void printArea();
}

class Triangle extends Shape{
void printArea(){
System.out.println("Area of triangle is: " + (double)(d1*d2)/2);
}
}

class Rectangle extends Shape{
void printArea(){
System.out.println("Area of rectangle is: " + (double)(d1*d2));
}
}


class Circle extends Shape{
void printArea(){
System.out.println("Area of circle: " + (double)(3.14*d1*d1));
}
}

class AreaMain{
public static void main(String args[]){
Rectangle r = new Rectangle();
Triangle tr = new Triangle();
Circle c = new Circle();
r.printArea();
tr.printArea();
c.printArea();
}
```

}

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.

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

• a)Accept deposit from customer and update the balance.

• b)Display the balance.

• c)Compute and deposit interest

• d)Permit withdrawal and update the balance

• Check for the minimum balance, impose penalty if necessary and update the balance.

```
import java.util.*;
import java.lang.Math;
abstract class Bank{
	abstract void withdraw(double amt);
	abstract void deposit(double amt);
	abstract void display();
	abstract void menudisp();
}

class Account extends Bank{
	String name;
	int acc_num;
	String type;
	double bal;
	String menu = " ";
	Account(String name, int acc_num, String type, double bal, String menu){
		this.name = name;
		this.acc_num = acc_num;
		this.type = type;
		this.bal = bal;
		this.menu = menu;
	}
	public void withdraw(double amt){
		if(amt>bal){
			System.out.println("Withdraw declined! Max amount you can withdraw is:
" + bal);
		}
```

```
                else{
                        bal -= amt;
                        System.out.println("Updated balance is: " + bal);
                }
        }

        public void deposit(double amt){
                bal += amt;
                System.out.println("Updated balance is: " + bal);
        }


        public void display(){
                System.out.println("Account number: " + name);
                System.out.println("Account name: " + acc_num);
                System.out.println("Account type: " + type);
                System.out.println("Balance: " + bal);
        }
        public void menudisp(){
                menu = "---------MENU---------\n1. Deposit\n2. Withdraw\n3. Display";
        }
}

class Savings extends Account{
        double interest;
        Savings(String name, int acc_num, String type, double bal,String menu, double interest){
                super(name, acc_num, type, bal,menu);
                this.interest = interest;
        }
        public double interest(int time){
                double comp;
                comp = bal + Math.pow((bal*(1+(interest/100))),time);
                return comp;
        }
        public void menudisp(){
                super.menudisp();
                menu += "\n4. Compute Interest\n5. Exit";
        }
}

class Current extends Account{
        double minbal = 10000;
        Current(String name, int acc_num, String type, double bal, String menu, double minbal){
                super(name, acc_num, type, bal,menu);
```

```
		this.minbal = minbal;
	}

	public void menudisp(){
		super.menudisp();
		menu += "\n4. Cheque Book\n5. Exit";
	}

	public void withdraw(double amt){
		if(amt>bal){
			System.out.println("Withdraw declined! Max amount you can withdraw is:
" + bal);
		}
		else{
			bal -= amt;
			System.out.println("Updated balance is: " + bal);
		}
	}

}

class BankMain{
	public static void main(String args[]){
		Scanner s = new Scanner(System.in);
		String name;
		String menu = " ";
		int acc_num;
		String type;
		double bal = 0;
		double interest;
		int choice;
		int time;
		double money;
		System.out.println("Enter customer name: ");
		name = s.next();
		System.out.println("Enter account number: ");
		acc_num = s.nextInt();
		System.out.println("---------ACCOUNT TYPE---------\n1.Savings
Account\n2.Current Account\nPlease select account type: ");
		type = s.next();

		if(type.equals("savings")){
			System.out.println("Enter interest amount: ");
			interest = s.nextDouble();
```

```
                        Savings accs = new Savings(name, acc_num, type, bal, menu, interest);
                        do{
                                accs.menudisp();
                                System.out.println(accs.menu);
                                System.out.println("Enter choice: ");
                                choice = s.nextInt();
                                switch(choice){
                                        case 1:
                                                System.out.println("Enter amount to be deposited:
");
                                                money = s.nextDouble();
                                                accs.deposit(money);
                                                break;
                                        case 2:
                                                System.out.println("Enter amount to be withdrawn:
");
                                                money = s.nextDouble();
                                                accs.withdraw(money);
                                                break;
                                        case 3:
                                                accs.display();
                                                break;
                                        case 4:
                                                System.out.println("Enter time to calculate interest
in years: ");
                                                time = s.nextInt();
                                                money = accs.interest(time);
                                                System.out.println("Compund interest is: " +
money);
                                                break;
                                        case 5:
                                                break;
                                }
                        }while(choice!=5);
                }

        }
}
```

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

 public void inputStudentDetails(){
  Scanner s=new Scanner(System.in);
  this.usn=s.nextLine();
  this.name=s.nextLine();
  this.sem=s.nextInt();
 }

 public void displayStudentDetails(){
  System.out.println(this.usn+" "+this.name+" "+this.sem);
 }
}
```

```
package CIE;
import java.util.Scanner;
public class Internals extends Student{
 protected int marks[]=new int[5];
 public void inputCIEmarks(){
  Scanner s=new Scanner(System.in);
  for(int i=0;i<5;i++){
   marks[i]=s.nextInt();
  }
 }
}
```

```
package SEE;
import CIE.Internals;
import java.util.Scanner;

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

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

 public void inputSEEmarks(){
  Scanner s = new Scanner(System.in);
  for(int i=0; i<5;i++){
   System.out.print("Subject "+(i+1)+" marks: ");
   marks[i] = s.nextInt();
  }
 }

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

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

}

import SEE.*;

class Main1{
 public static void main(String args[]){
  int num=2;
  Externals finalMarks[]=new Externals[num];
  for(int i=0;i<num;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<num;i++){
   finalMarks[i].calculateFinalMarks();
```

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


 }
}
```

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.

```
import java.util.*;
class WrongAge extends Exception{
    WrongAge(String s){
        super(s);
    }
}

class InputScanner{
    Scanner s = new Scanner(System.in);
}

class Father extends InputScanner{
    int fatherAge;
    Father() throws WrongAge {
        System.out.println("Enter father's age: ");
        fatherAge = s.nextInt();
        if (fatherAge < 0)
            throw new WrongAge("Age cannot be negative");
    }

    void displayf(){
        System.out.println("\nFather's  age: " + fatherAge);
    }
}

class Son extends Father{
    int sonAge;
    Son() throws WrongAge{
        super();
        System.out.println("Enter son's age: ");
```

```
        sonAge = s.nextInt();
        if (sonAge < 0)
            throw new WrongAge("Age cannot be negative.");
        else if(sonAge > fatherAge)
            throw new WrongAge("Son's age cannot be greater than father's age.");
        else{
            displayf();
            displaySon();
            throw new WrongAge("Valid age.");
        }
    }

    void displaySon(){
        System.out.println("\nSon's age : "+ sonAge);
    }
}

class exceptionsmain {
    public static void main(String[] args) {
        try{
            Son son = new Son();
        } catch(WrongAge e) {
            System.err.println(e.getMessage());
        }
    }

}
```

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.

```
import java.lang.*;
class DisplayMessageThread extends Thread {
    private final String message;
    private final long interval; // in milliseconds

    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 TwoThreadDemo {
    public static void main(String[] args) {
        DisplayMessageThread thread1 = new DisplayMessageThread("BMS College of
Engineering", 10000); // 10 seconds
        DisplayMessageThread thread2 = new DisplayMessageThread("CSE", 2000); // 2 seconds

        thread1.setName("Thread 1");
        thread2.setName("Thread 2");

        thread1.start();
        thread2.start();

        try {
            // Let the threads run for a while
            Thread.sleep(30000); // Let the program run for 30 seconds
        } catch (InterruptedException e) {
            System.out.println("Main thread interrupted.");
        }

        // Interrupt both threads to stop them
        thread1.interrupt();
        thread2.interrupt();

        System.out.println("Main thread exiting.");
    }
}
```

10)Demonstrate Inter process Communication and deadlock
a)Inter process Communication
b)Deadlock

```
import java.lang.*;
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();
            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 otherthread.

System.out.println("Back in otherthread");

}

public static void main(String args[]) {

new Deadlock();

}
```

```
  }
```

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 java.awt.*;
import java.awt.event.*;

public class DivisionMain1 extends Frame implements ActionListener
{
	TextField num1,num2;
	Button dResult;
	Label outResult;
	String out="";
	double resultNum;
	int flag=0;

	public DivisionMain1()
	{
		setLayout(new FlowLayout());

		dResult = new Button("RESULT");
		Label number1 = new Label("Number 1:",Label.RIGHT);
		Label number2 = new Label("Number 2:",Label.RIGHT);
		num1=new TextField(5);
		num2=new TextField(5);
		outResult = new Label("Result:",Label.RIGHT);

		add(number1);
		add(num1);
		add(number2);
		add(num2);
		add(dResult);
		add(outResult);

		num1.addActionListener(this);
		num2.addActionListener(this);
		dResult.addActionListener(this);
		addWindowListener(new WindowAdapter()
		{
```

```
                public void windowClosing(WindowEvent we)
                {
                        System.exit(0);
                }
        });
}

public void actionPerformed(ActionEvent ae)
{
        int n1,n2;
        try
        {
                if (ae.getSource() == dResult)
                {
                        n1=Integer.parseInt(num1.getText());
                        n2=Integer.parseInt(num2.getText());

                        /*if(n2==0)
                                throw new ArithmeticException();*/
                        out=n1+" "+n2;
                        resultNum=n1/n2;
                        out+=String.valueOf(resultNum);
                        repaint();

                }
        }
        catch(NumberFormatException e1)
        {
                flag=1;
                out="Number Format Exception!  "+e1;
                repaint();
        }
        catch(ArithmeticException e2)
        {
                flag=1;
                out="Divide by 0 Exception!  "+e2;
                repaint();
        }

}

public void paint(Graphics g)
{
        if(flag==0)
```

```
g.drawString(out,outResult.getX()+outResult.getWidth(),outResult.getY()+outResult.getHeight()-
8);
                else
                g.drawString(out,100,200);
                flag=0;
        }

        public static void main(String[] args)
        {
                DivisionMain1 dm=new DivisionMain1();
                dm.setSize(new Dimension(800,400));
                dm.setTitle("DivionOfIntegers");
                dm.setVisible(true);
        }

}
```