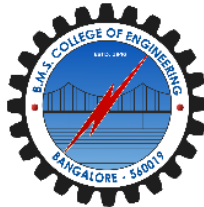# B.M.S. COLLEGE OF ENGINEERING

Basavanagudi, Bengaluru- 560019

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

LAB REPORT

On

Object-Oriented Java Programming

(23CS3PCOOJ)

Submitted By:

MOHAMMED SHURAIM

1BM22CS158

In partial fulfilment of

BACHELOR OF ENGINEERING

In

COMPUTER SCIENCE AND ENGINEERING

2023-24

B.M.S. COLLEGE OF ENGINEERING

Basavanagudi, Bengaluru- 560019

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Lab work entitled "Object-Oriented Java Programming (22CS3PCOOJ)" conducted by **MOHAMMED SHURAIM (1BM22CS158),** who is bonafide student at **B.M.S.College of Engineering**. It is in partial fulfilment for the award of **Bachelor of Engineering in Computer Science and Engineering** during the academic year 2023-24. The Lab report has been approved as it satisfies the academic requirements in respect of Object-Oriented Java Programming (23CS3PCOOJ) work prescribed for the said degree.

INDEX

PROGRAM 1

Develop a Java program that prints all real solutions to the quadratic equation ax2+bx+c= 0. Read in a, b, c and use the quadratic formula. If the discriminate b2-4ac is negative,display a message stating that there are no real solutions.

```java
import static java.lang.Math.*;
public class QuadraticEquation {
    static void calculateRoots(int a, int b, int c) {
        if (a == 0) {
            System.out.println("The value of a cannot be 0.");
            return;
        }
        int d = b * b - 4 * a * c;
        double sqrtval = sqrt(abs(d));
        if (d > 0) {
            System.out.println("The roots of the equation are real and different. \n");
            System.out.println((double)(-b + sqrtval) / (2 * a) + "\n"+ (double)(-b - sqrtval) / (2 * a));
        } else if (d == 0) {
            System.out.println("The roots of the equation are real and same. \n");
            System.out.println(-(double)b / (2 * a) + "\n"+ -(double)b / (2 * a));
        } else {
            System.out.println("The roots of the equation are complex and different. \n");
            System.out.println(-(double)b / (2 * a) + " + i"+ sqrtval + "\n"+ -(double)b / (2 * a)+ " - i" + sqrtval);
        }
    }
 public static void main(String args[]) {
        int a = 1, b = 5, c = 2;
                calculateRoots(a, b, c); //Output 2: int a=1, b=1, c=1
                                //Output 3: int a=1, b=-2, c=1
}
}
```

PROGRAM 1

OUTPUT :

int a = 1, b = 5, c = 2

```
The roots of the equation are real and different.

-0.4384471871911697
-4.561552812808831
```

int a=1, b=1, c=1

```
The roots of the equation are complex and different.

-0.5 + i1.7320508075688772
-0.5 - i1.7320508075688772
```

int a=1, b=-2, c=1

```
The roots of the equation are real and same.

1.0
1.0
```

Program 1

Write a java program to solve the quadratic equations i.e finding the roots

$$d = \sqrt{b^2 - 4ac}$$

d > 0	2 Solutions
d = 0	1 Solution
d < 0	2 Complex soln

```java
import java.util.Scanner;
import static java.lang.Math.*;
public class QuadEqn
{
    void calcRoots (int a, int b, int c)
    {
        if (a == 0)
        {
            System.out.println ("The value of a cannot
                be 0.");
            return;
        }

        int d = b*b - 4*a*c;
        double sqval = sqrt(abs(d));

        if (d > 0)
        {
            System.out.println (" The roots of the
            equation are real and different.\n");
            System.out.println ((double)(-b+sqval)/(2*a)
            +"\n"+(double(-b-sqval)/(2*a)));
        }
        else if (d == 0)
        {
            System.out.println ("The roots of the
            equation are real and same.\n");
            System.out.println (-(double)b/(2*a)+"\n"+
            (-(double) b/(2*a)));
        }
```

```java
else
{
System.out.println(" The roots of the equation
are complex and different.\n");

System.out.println(-(double)b/(2*a)+ "+i"
+ sqial + "\n" + -(double)b/(2*a)+ "-i"
+ sqial);

}
}

Public static void main (String[] args)
{
int x, y, z;

Scanner S2 = new Scanner(System.in);

System.out.println(" Enter the 3
coefficients ");
x = S2.nextInt();
y = S2.nextInt();
z = S2.nextInt();

CalcRoots(x,y,z);

}

}
```

7

```java
else
{
    System.out.println("The roots of the equation
    are complex and different. \n");

    System.out.println("- (double)b/(2*a)+ "+i"
    + sqval + "\n" + - (double)b/2*a)+"i"
    + sqval);

}
}

Public static void main (String[] )
{
    int x, y, z;

    Scanner S2 = newScanner(System.in);

    System.out.println("Enter the 3
    coefficients ");
    x = S2.nextInt();
    y = S2.nextInt();
    z = S2.nextInt();

    CalcRoots(x,y,z);

}

}
```

8

Output

Enter the coefficients of the equation

1

2

3

the roots of the equation are ~~real~~ complex and ~~same~~ different

$1.0 + i2.828427$

$-1.0 - i2.828427$

Enter the coefficients of the equation

1

5

2

The roots of the equation are real and different

$-0.438471871 9$

-4.56155812

Enter the coefficients of the equation

1

1

1

The roots of the equation are complex and different

$-0.5 + i1.73205080756$

$-0.5 - i11.7320508756$

PROGRAM 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.

```java
import java.util.*;
public class Main {
        Scanner sc=new Scanner(System.in);
        public int n;
        String name,usn;
         int marks[],credit[],sum;
        float sgpa;
        public void acceptAndDisplay(){

                System.out.println("enter number of subjects : ");
                n=sc.nextInt();
                sc.nextLine();
                credit=new int[n];
                marks=new int[n];

                System.out.println("enter your name :");
                name=sc.nextLine();
                System.out.println("enter your usn : ");
                usn=sc.nextLine();
        }
        void calculate(){
                for(int i=0;i<n;i++){
                        System.out.println("enter the credit points : ");
                        credit[i]=sc.nextInt();
                        System.out.println("enter the respective marks :");
                        marks[i]=sc.nextInt();
```

```java
                        sgpa+=credit[i]* marks[i]/10;

                        sum +=credit[i];

                }

                sgpa=sgpa/sum;

                System.out.println("sgpa is :"+sgpa);

        }

        public static void main(String[] args){


                Main m=new Main();

                m.acceptAndDisplay();

                m.calculate();

        }

}
```

PROGRAM 2

OUTPUT :

```
C:\Users\user> java Main
 enter number of subjects :
 3
 enter your name :
 John Doe
 enter your usn :
 123456
 enter the credit points :
 3
 enter the respective marks :
 85
 enter the credit points :
 4
 enter the respective marks :
 78
 enter the credit points :
 2
 enter the respective marks :
 92
 sgpa is : 8.04
```

Program2

Q write a java program to dalate SGPA.
import java.util.Scanner;
class Student {
String usn, name;
int n, credit[ ], marks[ ], sum;
float sgpa = 0.0;
Scanner sc = new Scanner (System.in)
void accept_and_display ()
{ System.out.println "00 en
System.out.println ("00 Enter Name:");
name = sc.next();
System.out.println ("00 Enter USN");
usn = sc.next();

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

System.out.println ("00 enter the
credits");
credits[i] = sc.nextInt();

System.out.println ("00 enter marks")
marks[i] = sc.nextInt();
}

void calculate ()
{
for (int i = 0; i < n; i++)
{
sgpa = sgpa + [credit[i] * (int) mark
/10;
sum = sum + credit[i]
}

```
sgpa = sgpa / sum;
System.out.println ("Sgpa is : " + sgpa);
}

Public static void main (String argf [])
{
    System.out.println ("enter no of subject")
    int n = s.nextInt();

    Student S = new Student ();
    S. Accept and Display ();
    S. Calculate ();

}
```

Enter the number of credits
8
Enter USN
1BM22CS157

enter the credits	enter marks
4	80
4	90
3	60
3	50
2	40
1	45
1	30
1	35
1	40

Sgpa : 9.6

PROGRAM 3

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

```java
import java.util.Scanner;

class Book {
    Scanner s1 = new Scanner(System.in);
    String name, author;
    int price, num_pages;

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

    Book() {}

    void accept() {
        System.out.println("Enter the name : ");
        name = s1.nextLine();
        System.out.println("Enter the name of the author : ");
        author = s1.nextLine();
        System.out.println("Enter the price of the book : ");
        price = s1.nextInt();
        System.out.println("Enter the number of pages : ");
        num_pages = s1.nextInt();
    }

    public String toString() {
```

```java
        return "Book Details : \n name : " + name + "\n author : " + author + "\n Price : " + price
                + "\n Number of pages : " + num_pages;
    }


    public static void main(String args[]) {
        int n;
        Scanner s = new Scanner(System.in);
        System.out.println("Enter the number of entries : ");
        n = s.nextInt();
        Book b[] = new Book[n];
        for (int i = 0; i < n; i++) {
            b[i] = new Book();
            b[i].accept();
        }
        for (int i = 0; i < n; i++) {
            System.out.println(b[i].toString());
        }
    }
}
```

PROGRAM 3

OUTPUT:

```
C:\Users\user> java Book
Enter the number of entries :
2
Enter the name :
The Great Gatsby
Enter the name of the author :
F. Scott Fitzgerald
Enter the price of the book :
15
Enter the number of pages :
180
Enter the name :
To Kill a Mockingbird
Enter the name of the author :
Harper Lee
Enter the price of the book :
12
Enter the number of pages :
281

Book Details :
 name : The Great Gatsby
 author : F. Scott Fitzgerald
 Price : 15
 Number of pages : 180
Book Details :
 name : To Kill a Mockingbird
 author : Harper Lee
 Price : 12
 Number of pages : 281
```

16

8/1/24

```java
import java.util.Scanner;
class Book {
    String name, author;
    int price, num_pages;
    Book (String n, String a, int p, int np) {
        this.name = n
        this.author = a;
        this.price = p; this.num_pages = np;
    }
    public String toString() {
        return "Book name:" + name + "\n
        author :" + author + "\n Price:"
        + Price + "\n Number of pages: " + num
        pages;
    }
}

public class B {
    static Scanner s = new Scanner(System.in)
    static Bookset()
    {
        s.nextLine();
        System.out.println("Enter book
        name");
        String n = s.nextLine();
        System.out.println("Enter author
                             name");
        String a = s.nextline();
        System.out.println("enter price
                of book");
        int p = s.nextint();
        System.out.println("enter no. of
                pages");
```

```
int np = S.nextInt();
Book b1 = new Book (n, a, p, np);
return b1;


Public static void main (String sx[])
{
    int n;
    System.out.println ("Enter no of
    books");
        n = S.nextInt();
        Book b[] = new Book[n]
        for (int i = 0; i < n; i++)
        b[i] = Set();
    System.out.println ("details
    of books entered");
        for (int i = 0; i < n; i++)
        System.out.println (b[i]);
    }
}
```

Q create a book class and create a
constructor Create get and set
method and make a method to
create an array of objects.
include tostring () method that could
display the complete details of the book.

Output: Enter no of books
2
Enter book name
ark

Enter author name
Noah

Enter price of book
456

Enter no. of pages
453

Enter book name
arkq
,
Enter author name
Hilla

Enter price of book
123

enter no. of pages
245

PROGRAM 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.

```java
import java.util.*;
abstract class shape
{
double a,b;
abstract void printArea(double a, double b);
}
class rectangle extends shape
{
void printArea(double a,double b)
{
double r;
r=a*b;
System.out.println("Area="+r);
}
}
class triangle extends shape
{
void printArea(double a, double b)
{
double r;
r=(a*b)/2;
System.out.println("Area="+r);
}
}
class circle extends shape
{
void printArea(double a, double b)
{
double r;
r=(a*a)*(3.14);
System.out.println("Area="+r);
}
}
class shapeDemo
{
public static void main(String args[])
{
Scanner sc=new Scanner(System.in);
int ch=1,d=1;
rectangle rec=new rectangle();
triangle tri=new triangle();
circle c=new circle();
```

```java
while(d==1)
{
System.out.println("Enter 1 for rectangle\n 2 for triangle \n 3 for circle\n 0 to exit");
ch=sc.nextInt();
switch(ch)
{
case 1:
System.out.println("Enter length and breadth");
double l=sc.nextDouble();
double b=sc.nextDouble();
rec.printArea(l,b);
break;
case 2:
System.out.println("Enter height and altitude");
double h=sc.nextDouble();
double a=sc.nextDouble();
tri.printArea(h,a);
break;
case 3:
System.out.println("Enter radius");
double r=sc.nextDouble();
c.printArea(r,3.14);
break;
case 0:
System.exit(1);
default:
System.out.println("wrong Input");
}
}
}
}
```

PROGRAM 4

OUTPUT :

```
C:\Users\user>java shapeDemo
Enter 1 for rectangle
 2 for triangle
 3 for circle
 0 to exit
1
Enter length and breadth
5
4
Area=20.0
```

```
C:\Users\user>java shapeDemo
Enter 1 for rectangle
 2 for triangle
 3 for circle
 0 to exit
2
Enter height and base
4
7
Area=14.0
```

```
C:\Users\user>java shapeDemo
Enter 1 for rectangle
 2 for triangle
 3 for circle
 0 to exit
3
Enter radius
3
Area=28.26
```

Program 4

Q) Develope a java program to create abstract class shape that contains two integers and an empty method named printArea(). Provide three classes rect, tri, cir, such that each of them extends to shape each class prints the method printArea() :

```java
import java.util.Scanner;
abstract class shape
{
    int x, y = 0;
    public shape (int x, int y)
    {
        this.x = x;
        this.y = y;
    }
    abstract void printarea();
}
class rect extend shape {
            public rect (int l, int b)
    {   supa(l,b); }
    public void printArea() {
System.out.println ("area of rectangle:"+(x*y));
} }
public void printarea()
{
    public rect (int l, int b)
    {
        super (b,h); }
    }
    public void printarea()
    { System.out.println ("area of triangle:"
    +(x*y*0.5)); }
```

```java
class cir extends shape {
    public cir ( int r)
    {
        super(r,r);
    }
    public void printarea() {
        System.out.println (" area of
        circle:" + (3.14 * x * x ));
    }
}
class shapes
{
    public static void main
    ( String [] args) {
    rect r1 = new rect ( 20,3);
    tri ti = new tri (5,10);
    cir c1 = new cir (4);
    r1.printarea ();
    t1.primarea ();
    c1.print area ();
    }
}
```

output
area of rectangle: 60
area of triangle : 25.0
area of circle : 50.24

PROGRAM 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.

```java
import java.util.Scanner;

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

    public Account(String customerName, int accountNumber, String accountType) {
        this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.accountType = accountType;
        this.balance = 0;
    }

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

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

class SavingsAccount extends Account {
    double interestRate;
```

```java
    public SavingsAccount(String customerName, int accountNumber) {
        super(customerName, accountNumber, "Savings");
        this.interestRate = 0.05; // 5% interest rate
    }

    public void depositInterest() {
        double interest = balance * interestRate;
        deposit(interest); `
        System.out.println("Interest deposited: " + interest);
    }

    public void withdraw(double amount) {
        if (balance >= amount) {
            balance -= amount;
            System.out.println("Withdrawal of " + amount + " successful.");
        } else {
            System.out.println("Insufficient balance for withdrawal.");
        }
    }
}

class CurrentAccount extends Account {
    double minimumBalance;
    double serviceCharge;

    public CurrentAccount(String customerName, int accountNumber) {
        super(customerName, accountNumber, "Current");
        this.minimumBalance = 1000; // Minimum balance required
        this.serviceCharge = 50; // Service charge if balance falls below minimum
    }

    public void withdraw(double amount) {
        if (balance - amount >= minimumBalance) {
            balance -= amount;
            System.out.println("Withdrawal of " + amount + " successful.");
        } else {
            System.out.println("Insufficient balance for withdrawal. Service charge of " +
serviceCharge + " will be applied.");
            balance -= serviceCharge;
        }
    }
}

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

        System.out.println("Enter name for savings account: ");
        String savingsName = scanner.nextLine();
        System.out.println("Enter account number for savings account: ");
        int savingsNumber = scanner.nextInt();
```

```java
        SavingsAccount savingsAccount = new SavingsAccount(savingsName,
savingsNumber);

        System.out.println("Enter name for current account: ");
        String currentName = scanner.nextLine(); // Consume newline
        currentName = scanner.nextLine();
        System.out.println("Enter account number for current account: ");
        int currentNumber = scanner.nextInt();
        CurrentAccount currentAccount = new CurrentAccount(currentName, currentNumber);

        System.out.println("Enter amount to deposit into savings account: ");
        double savingsDeposit = scanner.nextDouble();
        savingsAccount.deposit(savingsDeposit);
        savingsAccount.displayBalance();

        System.out.println("Enter amount to deposit into current account: ");
        double currentDeposit = scanner.nextDouble();
        currentAccount.deposit(currentDeposit);
        currentAccount.displayBalance();

        System.out.println("Enter amount to withdraw from savings account: ");
        double savingsWithdraw = scanner.nextDouble();
        savingsAccount.withdraw(savingsWithdraw);
        savingsAccount.displayBalance();

        System.out.println("Enter amount to withdraw from current account: ");
        double currentWithdraw = scanner.nextDouble();
        currentAccount.withdraw(currentWithdraw);
        currentAccount.displayBalance();
    }
}
```

PROGRAM 5

OUTPUT :

```
Enter name for savings account:
John Doe
Enter account number for savings account:
123456
Enter name for current account:
Jane Smith
Enter account number for current account:
654321
Enter amount to deposit into savings account:
5000
Deposit of 5000.0 successful.
Current balance: 5000.0
Enter amount to deposit into current account:
3000
Deposit of 3000.0 successful.
Current balance: 3000.0
Enter amount to withdraw from savings account:
2000
Withdrawal of 2000.0 successful.
Current balance: 3000.0
Enter amount to withdraw from current account:
4000
Insufficient balance for withdrawal. Service charge of 50.0 will be applied.
Current balance: 2950.0
```

Bank Program (5)

Develope a Java program to create
a class bank that maintains 2 kinds
of accounts for its customer. one called
savings account and he other current accau
The saving account provides compound intrest
and withdrawal facilites 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
th 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 derived
the class cur acct & savacct to make them
more specific to their requirements. Includ
the necessary methods in order to achiee
the following tasks:

a) Accept deposit from customer & update the
Dolance

6) display balance
c) compute & deposit interest
d) permit withdrawal & update to balanc

```
import java.util.Scanner;
class Account {
    String customerName;
    int accountnumber,
    String accountType;
    double balance;


    public Account ( String customerName
    int accountNumber, String accountType)
    {

    this. customerName = customerName
    this. accountNumber = accountNumber
    this. accountType = accountType
    this. balance = 0;
    }


    public void deposit (double amount)
    {
        balance += amount;
        System.out.println ("Deposit of " + account +
        "Successful.");
    }

    public void displayBalance()
    {
        System.out.println ("current balance" +
        balance);
    }

}

class SavingAccount extends Account {
    double interestRate;
    public SavingAccount ( String customerName,
    int accountNumber)
    { Super ( customerName, accountNumber, "Saving");
```

30

```java
this.interestRate = 0.05;
}

public void depositInterest()
{
    double interestInterest = balance * interestRa
    deposit(interest);
    System.out.println("Interest deposited."
    + interest);
}

public void withdraw(double amount)
{
    if(balance - amount >= minimumbalance)
    {
        balance -= amount;
        System.out.println("withdrawal of "
        + amount + " Successful");
    }
    else {
        System.out.println("Insufficient balance
        for withdrawal. service charge "
        + serviceCharge + " will be applied");
        balance -= serviceCharge;
    }
}
}
```

31

```java
class CurrentAccount extends Account {
    double minimumBalance;
    double serviceCharge;

    public CurrentAccount (String customerName,
    int accountNumber)
    {
        super (customerName, accountNumber, "current");
        this.minimumBalance = 1000;
        this.serviceCharge = 50;

    public void withdraw (double amount)
    {
        if (balance-amount >= minimum balance)
        {
            balance -= amount
            System.out.println ("withdrawal
            of" + amount + " successful.");
        else {
            System.out.println ("Insufficient balance
            for withdrawal. service charge of
            " + serviceCharge + " will be applied.");
            balance -= serviceCharge;
        }
    }
}
```

```java
public class Bank {
    public static void main (String a

        Scanner sc = new Scanner (System in);
        Saving account sa = new CurrAcc ("John
        Doe", 1001);
        SavingAcct SA = new SavAcc ("Jane
        Smith", 2001);

        CA.deposit (5000);
        SA.deposit (10000);
        CA.withdraw (2000);
        SA.withdraw (500);
        SA.displayBalance();
        SA.displayBalance();
        SA.CalculateInterest();
        SA.displayBalance();
```

Output:

```
deposit of 5000.0 Successful
current balance : 5006 0
deposit of 250.0 Successful
interest deposited :250.6
CurrentBalance: 5250 0
deposit of 200.0 successful
current balance: 300 0
Insufficient balance for withdrawal
Service charge of 50.0 will be applied
current balance 250. 0
```

PROGRAM 6

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

```
package CIE;


public class internals extends CIE.student{

        public double imarks[] ;

        public internals(String u, String n, int s ,double m[]) {

                super(u,n,s);

                this.imarks = m;

        }
}
```

```
package SEE;

import CIE.student;

public class externals extends CIE.student{

        public double smarks[];

        public externals (String u, String n, int s ,double m[]) {

                super(u,n,s);

                this.smarks = m;

        }
}
```

```
package result;

import CIE.student ;

import CIE.internals;

import SEE.externals;

public class test {

        public static void main(String[] args) {

                double internal[] ={43, 45, 47, 44, 41};

                double external[] = {90, 87, 65, 98, 43};

                student s1 = new student("1BM22CS158", "shuraim", 3);

                internals i1 = new internals("1BM22CS158", "shuraim", 3 ,internal);

                externals e1 = new externals("1BM22CS158", "shuraim", 3 , external);

                System.out.println("usn:" + s1.usn + " name: " + s1.name + " sem: "
+s1.sem);

                System.out.println("internal marks: ");

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

                        System.out.printf("internal marks %d : %f", i+1, i1.imarks[i]);

                }

                System.out.println();

                System.out.println("external marks: ");

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

                        System.out.printf("external marks %d : %f", i+1, e1.smarks[i]);

                }

        }

}
```

PROGRAM 6   OUTPUT :

```
usn:1BM22CS158 name: shuraim sem: 3
internal marks:
internal marks 1 : 43.000000internal marks 2 : 45.000000internal marks 3 : 47.000000 internal marks
444.000000internal marks 5 : 41.000000
external marks:
external marks:: 90.000000 external marks 2 : 87.000000 external marks 3 : 65.000000 external marks 4
 : 98.000000 external marks 5 : 43.000000
```

Program 6

Create a package CIE which has 2 classes- Student and Internals
The class personal has members like USN name sem, The class internal has an array that stores the Internal marks scored in 5 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 5 courses of the current semester of the student. Import the 2 packages in a file that declares that the final marks of n Students in all courses.

```java
Package cie;
public class student {
    public String usn;
    public String name;
    public int sem;
    public student (String v, String n, int s)
    {
        this.usn=v;
        this.name= n;
        this.sem= s;
    }
}
public class Internals extends cie.student
{
    public double imarks[];
    public Internals (String v, String n, int s,
    double m[]) { super (v, n, s);
```

```
        this.Pmarks = m;
     }
   }



Package SEE;
import CIE.students;
public class external extends
  CIE.student
  {
     public double smarks[];
     public externals (string v, string n,
   int s, double m[]) {
        super(v,n,s);
        this.Smarks = m;
     }
   }


Package result;
Import CIE.Students;
import CIE.internals;
import SEE.externals;
public class test {
public static void main (String
args[]) {
double internal[] = {43, 45, 47, 44, 41};
double external[] = {90, 87, 65, 98, 43};
Student s1 = new student ("IBM22CS150",
"Jose", 3);
  internal i1 = new internals ("IBM22
CS150", 3, internals);
external e1 = new external ("IBM22CS
150", 3, external);
```

```java
System.out.println ("USN:" + S1.USN +
"name: " + S1.name + " Sem:" + 2.sem);

    System.out.println ("Internals marks");
    for (i=0; i<5; i++)
    {
    System.out.println ("internal  marks"
    + (i+1) + i1.imarks[i]);
    }
    System.out.println ("external marks");
    for (int i=0; i<5; i++)
    {
    System.out.println ("External marks"
    + (i+1) + e1.smarks[i]);

        }
        }
    }
```

output

USN : IBM22CS150 Name: JOSE Sem:

internal mark1: 43.000 internal
marks2: 45.00 , internal mark3:
47.00 internal marks5: 41

External marks:
External marks : 90.000 external
mark 2 : 87.00 , marks4: 98.00

External marks5: 43.00

PROGRAM 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=father's age.

```java
class MyException extends Exception {
    int detail;
        public MyException(int age, String exe) {
                this.detail=age;
                System.out.println(exe + " given age is: " + age + " please enter again");
        }
        public String getMessage() {
    return "Exception: " + detail;
  }
}
class Father{
        int age;
        public Father(int age) throws MyException {
                if (age < 0) throw new MyException(age , "Age cannot be lesser than 0");
                this.age = age ;
        }
}
class Son extends Father{
        int age;
        public Son(int fatherAge, int sonAge) throws MyException{
                super(fatherAge);
                this.age = sonAge;
                if (this.age > super.age) throw new MyException(age , "Age of son cannot be
more than father");
```

```java
                }
        }
public class father_son {

        public static void main(String[] args) {

                try {

                        Father f1 = new Father(-1);

                        Son s1 = new Son(30 , 31);

                }

                catch (MyException e) {

                        System.out.println("Exception caught: " + e.getMessage());

                }

        }

}
```

PROGRAM 7

OUTPUT :

```
C:\Users\user>javac father_son.java
C:\Users\user>java father_son
Age cannot be lesser than 0 given age is: -1 please enter again
Exception caught: Exception: -1
Age of son cannot be more than father given age is: 31 please enter again
Exception caught: Exception: 31
```

Write a program that demonstrates
handling of exception in inheritance
tree. create a base class called
father and derieved class called
  Son which extends the base class.
In father class, implement a constructor
which take the age and throws the
expection wrongAge() when the i/p age
<0 on son class. Implement a
constructor thats takes both father
and Son's age and throws an
  exception if sons age >= father's age.

```java
class MyException extends Exception {
    int detail;
    public Myexception (int age, String exe)
    {
        this.detail = age;
        System.out.println (exe + "given age is:"
        + age + " please enter again");

    }
    public String getMessage() {
        return "exception: " + details;
    }
}


class Father {
int age;
public Father (int age) throws MyException
{
    if(age<0) throw new Myexception
    (age, " Age cannot be lessthan 0");
```

```java
        this.age = age;

    }
}

class Son extends Father {
    int age;
    public Son (int fatherAge, int SonAge)
    throws my Exception (age, "Age of
    Son cannot be more than father");
    }
}


public class Father_son {
    public static void main (String []
    args)
    {
    try {
        Father F1 = new Father (-1);
        Son S1 = new Son (30,31);
    }
    catch (myException e)
    {

        System.out.println ("exception
    caught " + e.getMessage();
    }
    }
}
```

:- input :

age cannot be lesser than 0 given
age is :- 1 plese enter again
exception caught Exception :- 1

PROGRAM 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 DisplayThread extends Thread {
    private String message;
    private int interval;

    public DisplayThread(String message, int interval) {
        this.message = message;
        this.interval = interval;
    }

    public void run() {
        while (true) {
            try {
                System.out.println(message);
                Thread.sleep(interval);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}

public class DisplayMessages {
    public static void main(String[] args) {
        Thread bmsThread = new DisplayThread("BMS College of Engineering", 10000); // 10
seconds
        Thread cseThread = new DisplayThread("CSE", 2000); // 2 seconds

        bmsThread.start();
        cseThread.start();
    }
}
```

PROGRAM 8

OUTPUT :

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

Program 8

write a program which execute
two threads are displaying `BMS
college of Engineering` once every
10 seconds and another displaying
"CSE" once every 2 seconds

```
class DisplayThread extends Thread
{
    private String message;
    private int interval;

    public DisplayThread (String message,
    int interval)
    {
        this.message = message;
        this.interval = interval;
    }

public void run ()
    {
    while (true)
        {
        try {

        System.out.println (message);
            Thread.sleep (interval);
        }
        catch (InterruptedException e)
        {
            e.printStackTrace ();
        }
        }
    }
```

```
class DisplayMessages
{
    public static void main (String [] args)
    {
        Thread bmsThread = new DisplayThread
        ("BMS College of Engineering," 10000);
        Thread cseThread = new DisplayThread
        ('CSE', 20000);

        bmsThread.start ();
        CSEThread.start ();
    }
}
```

Output
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE

PROGRAM 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.

```java
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);
```

```java
                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();


                    }

            }
```

```java
        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));
```
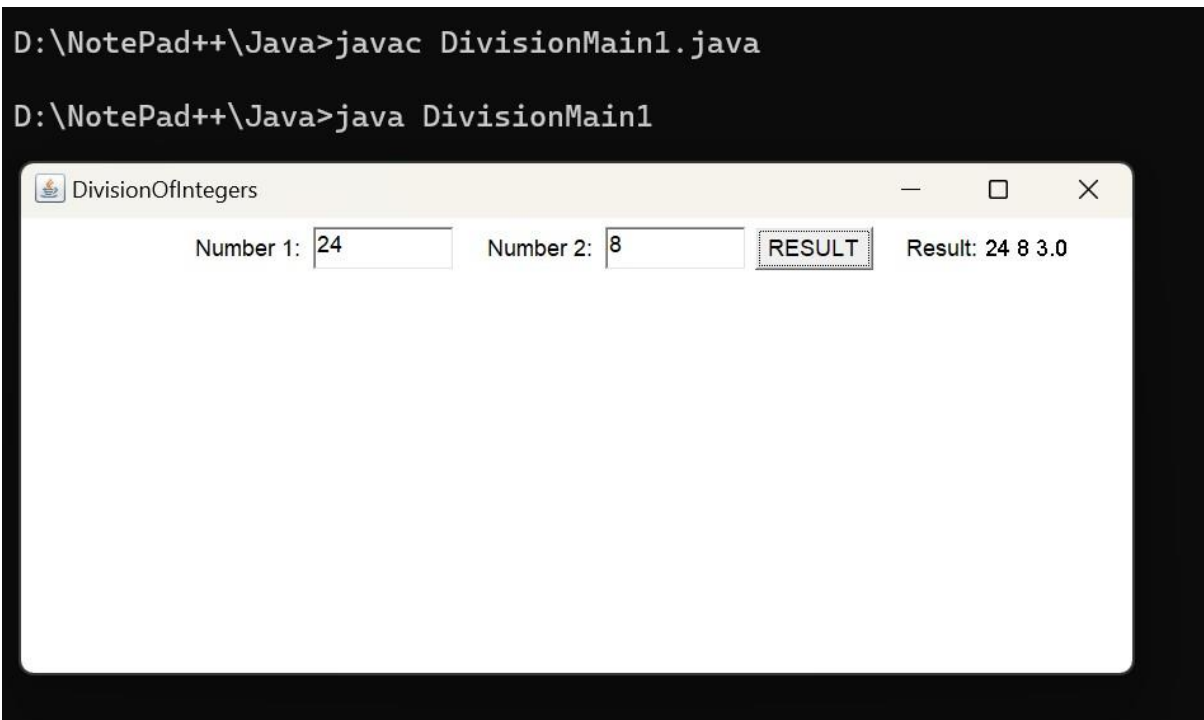
```
    dm.setTitle("DivisionOfIntegers");

    dm.setVisible(true);

    }

}
```

PROGRAM 9

OUTPUT :

Program 4

write a program that creates
a user interface to perform integer
divisions.
The user enter 2 no. in the text
fields, Num1 and num2. The division
of num1 & num2 is displayed in the
result field when the divide button
is clicked. If num1 or num2 was
not an integer, the program would
throw a numberFormatException. If
num2 were zero, the program would
throw an arithmetic exception display
the exception in a message dialog box.

```java
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 ("60 number:",
Label. RIGHT);
num1 = new TextField (5);
num2 = new TextField (5);
outResult = new Label ("60 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 Arithmetic
    Exception(); */
```

```java
out = n1 + " " + n2 + "";
resultNum = n1 / n2;
out + = String. value of (result Num);
repaint ();
}
}
catch ( NumberFormat Exception e1)
{
    flag = 1;
    out = " Number Format Exception! " + e1;
    repaint ();
}
catch (Arithmetic Exception e2 )
{
    flag = 1;
    out = " Divide by 0 Exception! " + e2;
    repaint ();
}
}
public void paint (Graphics g)
{
    if (flag == 0)
    g.drawString (out, outResult. getX()
    + outResult. get width (), outResult
    . getY() + outResult. getHeight () - 8);
    else
    g. draw String (out, 100, 200);
    flag = 0;
}
```

```java
Public static void main (String []
args)
{
DivisionMain1 ();  dm.setSize (
new Demension (800, 400 ));
dm. setTitle ("Division Of Integers");
dm. setVisible (true);
}
```

Output

Division Of Integers   — □ ×

number1 [24]   Number2: [8]   [Result]

                        Result : 24830

20/2/24

program 9 (Report)

Nepaol

The given program utilizes
Java's AWT and swing libraries
to create GUI Applications. These
programs showcase various event
handling in Java.

(i) Button Demo: Its an applet that
demonstrates event handling in Java
AWT. It consists of three buttons
labeled 'yes', 'no' and 'undecided".
clicking on each button triggers
an action event. And the corresponding
message is displayed on the applet

(ii) Button List: Its another frame
based Java application that
demonstrates event handling
and consist of three buttons similar
to Button Demo program. clicking on
any button updates a message
indicating the button's pressed.

(iii) buttondrag: it is a frame-based
Java app that implements a puzzle
game. there players rearrange
numbered buttons in ascending
orders by swapping their position.

Report on MouseEventsDemo program

This showcase the implementation of mouse event handling in Java, It provides a simple graphical user interface where user can interact with the mouse & the program responds to various mouse event.

In conclusion this program through its intuitive interface & dynamic feedback users can understand & interact with different mouse action.

iv) DivisionMain : It is a frame-based
Java app that allows users to input
2 numbers and calculate their division.
It includes error handling for
scenarios such as division by 0 and
invalid input formats.

v) Division Main1 : Its another frame
based Java app that performs
divisions operation similar to
DivisionMain. However it handles
exceptions in a different order compared
to DivisionMain.

vi) Text Field Demo: Demonstrates the usage of
the text fields in Java AWT. It
provides a simple GUI interfaces
where user can input their name. and
password. Upon pressing enter in either
text file. the program repaints the
window to display the entered name
and password.

vii) Button dist1): Is a frame based
Java app that demonstrates event
handling and dialog creation. It
consist of 3 button labeled "yes, "no'
and "undecided". clicking on any
button opens a analog window displaying
the button label.

In conclusion the mentioned Java
programs exemplify fundamental GUI
development & event handling techniques.