VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“Jnana Sangama”, Belgaum -590014, Karnataka.

PROJECT WORK REPORT

On

“OOJ LAB REPORT”

Submitted by

ARIN DSOUZA(1BM22CS052)

Under the Guidance of

SONIKA MAAM

ASSISTANT PROFESSOR

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

B.M.S. COLLEGE OF ENGINEERING

(Autonomous Institution under VTU)

BENGALURU-560019

Mar 2024

B. M. S. College of Engineering,

Bull Temple Road, Bangalore 560019

(Affiliated To Visvesvaraya Technological University, Belgaum)

Department of Computer Science and Engineering

B.M.S. COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

I, **ARIN DSOUZA (1BM22CS052)** of 3rd Semester, B.E, Department of Computer Science and Engineering, BMS College of Engineering, Bangalore, hereby declare that, this PROJECT entitled **"OOJ LAB REPORT"** has been carried out by me under the guidance of Sonika ma'am, Assistant Professor, Department of CSE, BMS College of Engineering, Bangalore during the academic semester Dec 2023 - Mar 2024.

We also declare that to the best of our knowledge and belief, the development reported here is not from part of any other report by any other students.

Signature

ARIN DSOUZA(1BM22CS052)

TABLE OF CONTENTS

3	**Week 3-** **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. **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 contains only the method printArea( ) that prints the area of the given shape.	17-23

4	**Week 4-** **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.	23-35
5	Week 5 – **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	19-21

	file that declares the final marks of n students in all five courses.	
6	Week 6 – **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<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. **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.	22-2
7	Week 7 – **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.	27-30

WEEK 1

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 discriminant b2-4ac is negative, display a message stating that there are no real solutions

Code:

```
import java.util.*;
public class QuadEq
{
   public static void main(String args[])
   {
      float a, b, c, d=0.0f, r1=0.0f, r2=0.0f;
      System.out.println("Enter values of a, b, c: ");
      Scanner read= new Scanner(System.in);
      a=read.nextFloat();
      b=read.nextFloat();
      c=read.nextFloat();
      if(a==0||b==0||c==0)
      {
         System.out.println("Invalid Input");
      }
      else
      {
         d=b*b-4*a*c;
         if(d>0)
         {
            r1=(float)(-b+Math.sqrt(d))/(2*a);
            r2=(float)(-b-Math.sqrt(d))/(2*a);
```

```
            System.out.println("Roots are real and distinct\nR1= "+r1+"\tR2= "+r2);
        }
        else if(d<0)
        {
            System.out.println("Roots are imaginary");
        }
        else
        {
            r1=-b/(2*a);
            r2=r1;
            System.out.println("Roots are real and equal\nR1= "+r1+"\tR2= "+r2);
        }
    }
    System.out.println("Name: Arin Dsouza \nUSN: 1BM22CS052");
  }
}
```

OUTPUT:

```
C:\Users\Arin\Desktop\oojreport>java QuadEq
Enter values of a, b, c:
6 7 6
Roots are imaginary
Name: Arin Dsouza
USN: 1BM22CS052
```

```
C:\Users\Arin\Desktop\oojreport>java QuadEq
Enter values of a, b, c:
1 2 1
Roots are real and equal
R1= -1.0        R2= -1.0
Name: Arin Dsouza
USN: 1BM22CS052
```

```
C:\Users\Arin\Desktop\oojreport>java QuadEq
Enter values of a, b, c:
1 6 1
Roots are real and distinct
R1= -0.17157288 R2= -5.8284273
Name: Arin Dsouza
USN: 1BM22CS052
```

\

WEEK 2

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.

Code:

```
import java.util.*;
public class Student {
    String name, usn;
    int credits[], marks[];
    public void display(double res) {
        System.out.println("Name : " + name);
        System.out.println("USN : " + usn);
        for (int i = 0; i < credits.length; i++) {
            System.out.println("Subject " + (i + 1) + " :\t Marks= " + marks[i] + "\tCredits= " +
credits[i]);
        }
        System.out.println("\nSGPA : " + res);
    }
    public double sgpa() {
        double tc = 0;
        double tgp = 0;
        for (int i = 0; i < credits.length; i++) {
            tc += credits[i];
            tgp += calgp(marks[i]) * credits[i];
        }
        return (tgp / tc);
    }
    public double calgp(int m) {
        if (m >= 90) {
```

```
            return 10;
        } else if (m >= 80) {
            return 9;
        } else if (m >= 70) {
            return 8;
        } else if (m >= 60) {
            return 7;
        } else if (m >= 50) {
            return 6;
        } else {
            return 0;
        }
    }
    public static void main(String args[]) {
        Scanner read = new Scanner(System.in);
        Student ob = new Student();

        System.out.print("\nEnter name: ");
        ob.name = read.nextLine();

        System.out.print("\nEnter USN: ");
        ob.usn = read.next();

        System.out.print("\nEnter no. of subjects: ");
        int n = read.nextInt();
        ob.credits = new int[n];
        ob.marks = new int[n];

        System.out.println("Enter marks and credits:");
        for (int i = 0; i < n; i++) {
            System.out.println("Marks for subject " + (i + 1) + ": ");
            ob.marks[i] = read.nextInt();
            System.out.println("Credits for subject " + (i + 1) + ": ");
            ob.credits[i] = read.nextInt();
        }

        double res = ob.sgpa();
        ob.display(res);
        System.out.println("Name: Arin Dsouza \nUSN: 1BM22CS052");
    }
}
```

OUTPUT:

```
C:\Users\Arin\Desktop\oojreport>java Student

Enter name: aryan

Enter USN: 1BM22CS055

Enter no. of subjects: 4
Enter marks and credits:
Marks for subject 1:
98
Credits for subject 1:
4
Marks for subject 2:
96
Credits for subject 2:
4
Marks for subject 3:
86
Credits for subject 3:
3
Marks for subject 4:
89
Credits for subject 4:
3
Name : aryan
USN : 1BM22CS055
Subject 1 :      Marks= 98      Credits= 4
Subject 2 :      Marks= 96      Credits= 4
Subject 3 :      Marks= 86      Credits= 3
Subject 4 :      Marks= 89      Credits= 3

SGPA : 9.571428571428571
Name: Arin Dsouza
USN: 1BM22CS052
```

WEEK 3

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.

Code:

```
import java.util.*;


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


    books() {
    }


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


    void getdata() {
        System.out.println("Enter the name of the book");
        name = sc.nextLine();
```

```
        System.out.println("Enter the name of the author");
        author = sc.nextLine();
        System.out.println("Enter the price");
        price = sc.nextInt();
        System.out.println("Enter the number of pages");
        num_pages = sc.nextInt();
    }

    public String toString() {
        return ("Name : " + name + "\nAuthor : " + author + "\nPrice : " + price + "\nNumber of
pages : " + num_pages);
    }
}

class bookdetails {
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        books b1 = new books("Mehta", "neil", 56, 154);
        System.out.println("Constructor values : \n" + b1);
        System.out.println("Enter the number of object of books");
        int n = sc.nextInt();
        books s[] = new books[n];
        for (int i = 0; i < n; i++) {
            s[i] = new books();
            System.out.println("Enter the details of " + (i + 1) + "book");
            s[i].getdata();
```

```
        }
        for (int i = 0; i < n; i++) {
            System.out.println("\nDetails of the book" + (i + 1));
            System.out.println(s[i]);
        }
        System.out.println("Arin");
        System.out.println("1BM22CS052");
    }
}
```

OUTPUT:

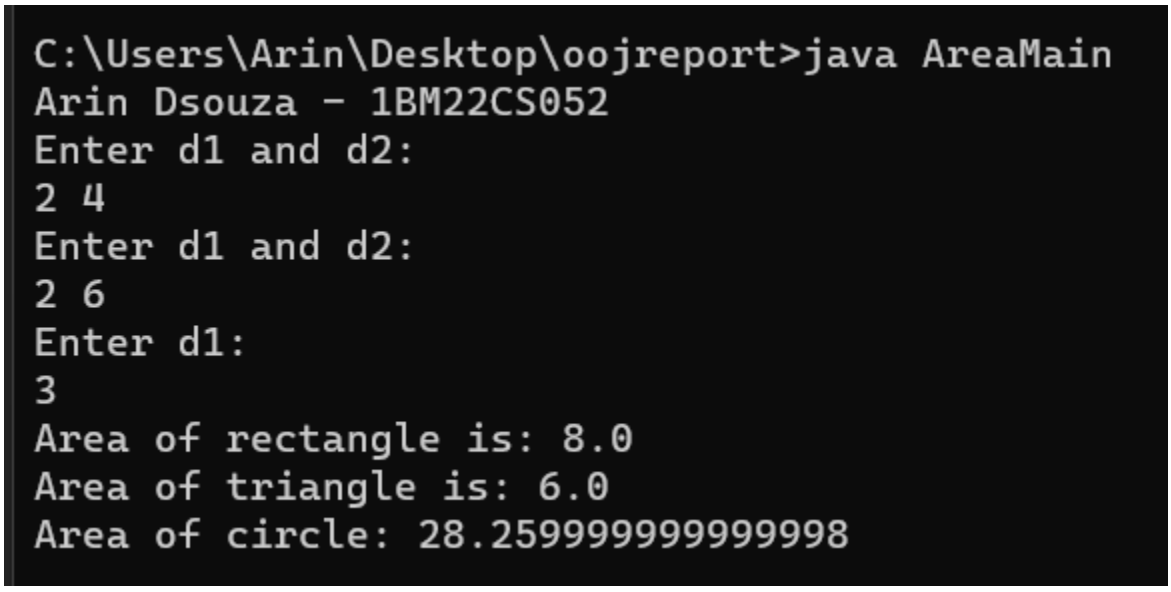
```
C:\Users\Arin\Desktop\oojreport>java bookdetails
Constructor values :
Name : Mehta
Author : neil
Price : 56
Number of pages : 154
Enter the number of object of books
1
Enter the details of 1book
Enter the name of the book
Time never stops
Enter the name of the author
steph curry
Enter the price
599
Enter the number of pages
400

Details of the book1
Name : Time never stops
Author : steph curry
Price : 599
Number of pages : 400
Arin
1BM22CS052
```

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 contains only the method printArea( ) that prints the area of the given shape.

Code:

```
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 d1: ");
d1 = sc.nextInt();
}
else{
System.out.println("Enter d1 and d2: ");
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[]){
System.out.println("Arin Dsouza - 1BM22CS052");
Rectangle r = new Rectangle();
Triangle tr = new Triangle();
Circle c = new Circle();
r.printArea();
tr.printArea();
c.printArea();
}
}
```



WEEK 4

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.

Code:

```
import java.util.Scanner;


class Account {

   String customerName;

   int accountNumber;

   String accountType;

   double balance;


   Account(String name, int number, String type, double initialBalance) {

      customerName = name;

      accountNumber = number;

      accountType = type;
```

```
    balance = initialBalance;
}

void deposit(double amount) {
    if (amount > 0) {
        balance += amount;
        System.out.println("Deposit of INR " + amount + " successful");
    } else {
        System.out.println("Invalid deposit amount. Please enter a positive value.");
    }
}

void displayBalance() {
    System.out.println("Account Number: " + accountNumber);
    System.out.println("Customer Name: " + customerName);
    System.out.println("Account Type: " + accountType);
    System.out.println("Balance: INR " + balance);
}

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

```
  void computeInterest() {
 }
    void checkMinimumBalance(double minBalance, double serviceCharge) {
    }
}

class SavAcct extends Account {
    double interestRate = 0.05;

    SavAcct(String name, int number, String type, double initialBalance) {
        super(name, number, type, initialBalance);
    }

    void computeInterest() {
        double interest = balance * interestRate;
        balance += interest;
        System.out.println("Interest of INR " + interest + " added to the account");
    }
}

class CurAcct extends Account {
    double minBalance = 1000;
    double serviceCharge = 50;

    CurAcct(String name, int number, String type, double initialBalance) {
        super(name, number, type, initialBalance);
    }
```

```
    void checkMinimumBalance(double minBalance, double serviceCharge) {
        if (balance < minBalance) {
            System.out.println("Service charge of INR " + serviceCharge + " imposed");
            balance -= serviceCharge;
        }
    }
}

public class Bank {
    public static void main(String[] args) {
        System.out.println("Arin Dsouza");
        System.out.println("1BM22CS052");
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of users: ");
        int numUsers = scanner.nextInt();

        Account[] accounts = new Account[numUsers];
        for (int i = 0; i < numUsers; i++) {
            System.out.println("\nUser " + (i + 1));
            System.out.print("Enter customer name: ");
            scanner.nextLine();
            String name = scanner.nextLine();
            System.out.print("Enter account number: ");
            int accNumber = scanner.nextInt();
            System.out.print("Enter initial deposit amount: INR ");
            double initialDeposit = scanner.nextDouble();
```

```
    System.out.print("Enter account type (Savings/Current): ");
    scanner.nextLine();
    String accType = scanner.nextLine();
    if (accType.equalsIgnoreCase("Savings")) {
        accounts[i] = new SavAcct(name, accNumber, accType, initialDeposit);
    } else if (accType.equalsIgnoreCase("Current")) {
        accounts[i] = new CurAcct(name, accNumber, accType, initialDeposit);
    } else {
        System.out.println("Invalid account type entered. Defaulting to Account.");
        accounts[i] = new Account(name, accNumber, "Account", initialDeposit);
    }
}

boolean exit = false;
while (!exit) {
    System.out.println("\nChoose an option:");
    System.out.println("1. Deposit");
    System.out.println("2. Withdraw");
    System.out.println("3. Display Balance");
    System.out.println("4. Compute Interest (Savings only)");
    System.out.println("5. Exit");
    System.out.print("Enter your choice: ");
    while (!scanner.hasNextInt()) {
        System.out.println("Invalid input. Please enter a number.");
        scanner.next();
    }
```

```
int choice = scanner.nextInt();
switch (choice) {
    case 1:
        System.out.print("Enter account number: ");
        int accNum = scanner.nextInt();
        System.out.print("Enter deposit amount: INR ");
        double depositAmount = scanner.nextDouble();
        for (Account acc : accounts) {
            if (acc.accountNumber == accNum) {
                acc.deposit(depositAmount);
            }
        }
        break;
    case 2:
        System.out.print("Enter account number: ");
        accNum = scanner.nextInt();
        System.out.print("Enter withdrawal amount: INR ");
        double withdrawAmount = scanner.nextDouble();
        for (Account acc : accounts) {
            if (acc.accountNumber == accNum) {
                acc.withdraw(withdrawAmount);
            }
        }
        break;
    case 3:
```

```
            System.out.print("Enter account number: ");
            accNum = scanner.nextInt();
            for (Account acc : accounts) {
                if (acc.accountNumber == accNum) {
                    acc.displayBalance();
                }
            }
            break;
        case 4:
            System.out.print("Enter account number (for Savings account): ");
            accNum = scanner.nextInt();
            for (Account acc : accounts) {
                if (acc.accountNumber == accNum && acc instanceof SavAcct) {
                    ((SavAcct) acc).computeInterest();
                }
            }
            break;
        case 5:
            exit = true;
            break;
        default:
            System.out.println("Invalid choice. Please enter a valid option.");
    }
}
```

OUTPUT:

```
C:\Users\Arin\Desktop\oojreport>java Bank
Name : Arin Dsouza
 USN:1BM22CS052
Enter the number of users: 1

User 1
Enter customer name: joshua
Enter account number: 12345
Enter initial deposit amount: INR 7000
Enter account type (Savings/Current): savings

Choose an option:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings only)
5. Exit
Enter your choice: 2
Enter account number: 12345
Enter withdrawal amount: INR 500
Withdrawal of INR 500.0 successful

Choose an option:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings only)
5. Exit
Enter your choice: 3
Enter account number: 12345
Account Number: 12345
Customer Name: joshua
Account Type: savings
Balance: INR 6500.0

Choose an option:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings only)
5. Exit
Enter your choice: 4
Enter account number (for Savings account): 12345
Interest of INR 325.0 added to the account

Choose an option:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings only)
5. Exit
Enter your choice: 5
```

```
C:\Users\Arin\Desktop\oojreport>java Bank
Name : Arin Dsouza
 USN:1BM22CS052
Enter the number of users: 1

User 1
Enter customer name: joshua
Enter account number: 12345
Enter initial deposit amount: INR 7000
Enter account type (Savings/Current): Current

Choose an option:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings only)
5. Exit
Enter your choice: 1
Enter account number: 12345
Enter deposit amount: INR 500
Deposit of INR 500.0 successful

Choose an option:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings only)
5. Exit
Enter your choice: 2
Enter account number: 12345
Enter withdrawal amount: INR 1000
Withdrawal of INR 1000.0 successful

Choose an option:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings only)
5. Exit
Enter your choice: 3
Enter account number: 12345
Account Number: 12345
Customer Name: joshua
Account Type: Current
Balance: INR 6500.0

Choose an option:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings only)
5. Exit
Enter your choice: 4
Enter account number (for Savings account): 12345

Choose an option:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings only)
5. Exit
Enter your choice: 5

C:\Users\Arin\Desktop\oojreport>
```

WEEK 5

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.

CODE:

File 1: Main.java

```
import SEE.Externals;
class Main{
public static void main(String args[]){
int numOfStudents=2;
Externals finalMarks[]=new Externals[numOfStudents];
for(int i=0;i<numOfStudents;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<numOfStudents;i++){
finalMarks[i].calculateFinalMarks();
finalMarks[i].displayFinalMarks();
}
}
}
```

File 2: Internals.java (Inside package CIE)

```
package CIE;
import java.util.Scanner;
public class Internals extends Student{
protected int marks[]=new int[5];
public void inputCIEmarks(){
```

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

File 3: Student.java (Inside package CIE)

```
package CIE;
import java.util.Scanner;
public class Student{
protected String usn=new String();
protected String name=new String();
protected int sem;
public void inputStudentDetails(){

System.out.println("Enter Details of students:");
Scanner s=new Scanner(System.in);
System.out.println("Enter USN:");
usn=s.nextLine();
System.out.println("Enter Name:");
name=s.nextLine();
System.out.println("Enter Semester:");
sem=s.nextInt();
}
public void displayStudentDetails(){
System.out.println("USN: "+usn);
System.out.println("Name: "+name);
System.out.println("Semester: "+sem);
}
}
```

File 4: Externals.java (Inside package SEE)

```
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]);
}
}
```

OUTPUT:

```
50
Enter the marks for subject 5:
50
Enter SEE marks
Subject1 marks:50
Subject2 marks:50
Subject3 marks:50
Subject4 marks:50
Subject5 marks:50
Displaying data:

USN: 1BM22CS052
Name: Arin Dsouza
Semester: 3
Subject1: 75
Subject2: 75
Subject3: 75
Subject4: 75
Subject5: 75
USN: AppleBee
Name: Alpha
Semester: 3
Subject1: 75
Subject2: 75
Subject3: 75
Subject4: 75
Subject5: 75
```

WEEK 6

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<0. In Son class, implement a constructor that cases both father and son’s age and throws an exception if son’s age is >=father’s age.

CODE:

```
import java.util.*;
class WrongAge extends Exception
{
	String message;
	public WrongAge(String msg)
	{
		this.message=msg;
		System.out.println(msg);
	}
}
class Father
{
	int f_age;
	Father(int f_age) throws WrongAge
	{
		if(f_age<0)
		{
			throw new WrongAge("Age cant be less than 0");
		}
		this.f_age=f_age;
}
}
class Son extends Father
{
	int s_age;
	Son(int f_age,int s_age) throws WrongAge
	{
		super(f_age);
		if(f_age<=s_age)
```

```
                {
                throw new WrongAge("Father can't be younger than son");
                }
                this.s_age=s_age;
}
}
class AgeTest
{
        public static void main(String args[])
        {
                int f,s;
                Scanner input=new Scanner(System.in);
                System.out.println("Enter age of father and son\n");
                f=input.nextInt();
                s=input.nextInt();
                try{
                        Father ob1=new Father(f);
                        Son ob2=new Son(f,s);
                }
                catch(WrongAge e)
                {
                        System.out.println("Caught");
                }
        System.out.println("Name: Arin Dsouza\nUSN: 1BM22CS052");
        }
}
```

OUTPUT:

```
C:\Users\Arin\Desktop\oojreport>java AgeTest
Enter age of father and son

50
19
Name: Arin Dsouza
USN: 1BM22CS052

C:\Users\Arin\Desktop\oojreport>java AgeTest
Enter age of father and son

-10
30
Age cant be less than 0
Caught
Name: Arin Dsouza
USN: 1BM22CS052

C:\Users\Arin\Desktop\oojreport>javac AgeTest
error: Class names, 'AgeTest', are only accepted if annotation processing is explicitly requested
1 error

C:\Users\Arin\Desktop\oojreport>java AgeTest
Enter age of father and son

20
30
Father can't be younger than son
Caught
Name: Arin Dsouza
USN: 1BM22CS052
```

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.

CODE:

```
class DisplayThread extends Thread {
    private String message;
    private int intervalMillis;
    public DisplayThread(String message, int intervalMillis) {
        this.message = message;
        this.intervalMillis = intervalMillis;
    }
    public void run() {
        while (true) {
            try {
```

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

public class Demo {
   public static void main(String[] args) {
      DisplayThread thread1 = new DisplayThread("BMS College of Engineering", 10000); // 10
seconds
      DisplayThread thread2 = new DisplayThread("CSE", 2000); // 2 seconds
      thread1.start();
      thread2.start();
System.out.println("Name:Arin Dsouza \n USN:1BM22CS052");
   }
```

OUTPUT:

```
C:\Users\Arin\Desktop\oojreport>java Demo
Name:Arin Dsouza
 USN:1BM22CS052
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
```

WEEK 7

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.

Code:

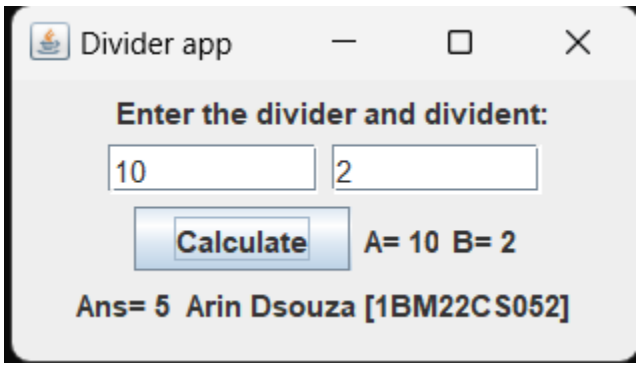
```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
class SwingDemo
{
   SwingDemo()
   {
      JFrame jfrm= new JFrame("Divider app");
      jfrm.setSize(265,150);
      jfrm.setLayout(new FlowLayout());
      jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      JLabel jlab=new JLabel("Enter the divider and divident: ");
      JTextField ajtf=new JTextField(8);
      JTextField bjtf=new JTextField(8);

      JButton button = new JButton("Calculate");

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

      //add in order
      jfrm.add(err);//to display error
      jfrm.add(jlab);
      jfrm.add(ajtf);
      jfrm.add(bjtf);
      jfrm.add(button);
      jfrm.add(alab);
      jfrm.add(blab);
 jfrm.add(anslab);

      ActionListener I =new ActionListener()
      {
         public void actionPerformed(ActionEvent evt)
         {
            System.out.println("Action event from a text field");
         }
      };
      ajtf.addActionListener(I);
```

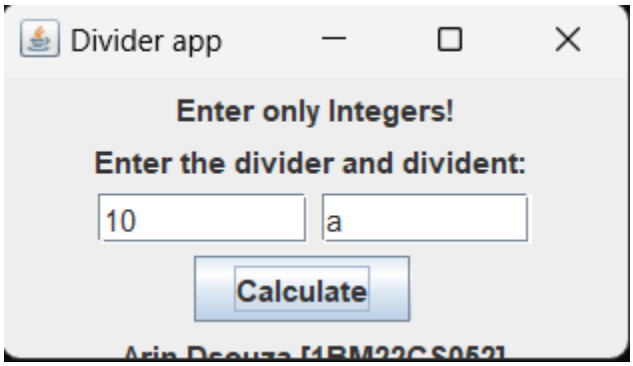
```
    bjtf.addActionListener(I);

    button.addActionListener(new ActionListener()
    {
       public void actionPerformed(ActionEvent evt)
       {
          if (err.getText()!=null)
             err.setText("");
       try
       {
          int a =Integer.parseInt(ajtf.getText());
          int b =Integer.parseInt(bjtf.getText());
          int ans= a/b;

          alab.setText("\nA= "+a);
          blab.setText("\nB= "+b);
          anslab.setText("\nAns= "+ans+"  Arugunta Hamsika [1BM22CS054]");
       }
       catch(NumberFormatException e)
       {
          alab.setText("");
          blab.setText("");
          anslab.setText("Arugunta Hamsika [1BM22CS054]");

          err.setText("Enter only Integers!");
       }
       catch(ArithmeticException e)
       {
          alab.setText("");
          blab.setText("");
          anslab.setText("Arugunta Hamsika [1BM22CS054]");
    err.setText("B should be NON zero!");
       }
    }
    });

    //display frame
    jfrm.setVisible(true);
}
```

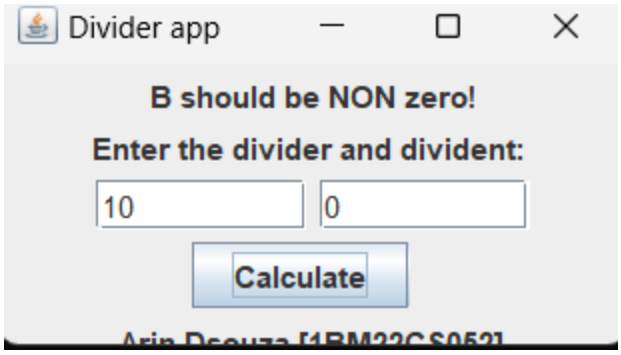
```
    public static void main(String args[])
    {
        //create frame on event dispatching thread
        SwingUtilities.invokeLater(new Runnable()
        {
            public void run()
            {
                new SwingDemo();
            }
        });
    }
}
```

OUTPUT:

```
C:\Users\Arin\Desktop\oojreport>javac SwingDemo.java

C:\Users\Arin\Desktop\oojreport>java SwingDemo
```






Divider app
B should be NON zero!
Enter the divider and divident:
10
0
Calculate