VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT

on

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted by

V Jayanth Vikram (1BM23IS275)

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

B.M.S. College of Engineering,

Bull Temple Road, Bangalore 560019

(Affiliated To Visvesvaraya Technological University, Belgaum)

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the Lab work entitled **Object Oriented Java Programming** carried out by **V Jayanth Vikram (1BM23IS275),** who is bonafide student of **B.M.S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum. The Lab report has been approved as it satisfies the academic requirements in respect of aObject Oriented Programming in Java(23CSPCOOJ) work prescribed for the said degree.

Lab faculty Incharge Name Assistant Professor Department of CSE, BMSCE	Swathi Sridharan Assistant Professor Department of CSE, BMSCE

Index

Sl. No.	Date	Experiment Title	Page No.
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			

Github Link: https://github.com/JVikramV/Java-Programs

Program 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.

Code:

```
import java.util.Scanner;

public class quad {
   public static void main(String[] args) {
   int a,b,c;
   float disc;
Scanner input = new Scanner(System.in);
System.out.println("Enter co-efficient of x square");
a= input.nextInt();
System.out.println("Enter co-efficient of x ");
b= input.nextInt();
System.out.println("Enter the constant");
c=input.nextInt();
disc = b*b-4*a*c;
if(disc<0){
System.out.println("No real root exists");
}

else if(disc>0){
double root1=(-b+Math.sqrt(disc))/(2*a);
double root2=(-b-Math.sqrt(disc))/(2*a);
System.out.println("The roots are" +root1+"and"+root2);
}

else{
double root1 = (-b)/(2*a);
System.out.println(" The roots are equal " + root1);
}

input.close();

}}
```

Date __/__/__
Page _____

Program-1

Q.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.

A.1)
```
import java.util.Scanner;
public class Quad
{
    public static void main(String[] args)
    {
        Scanner scanner = new Scanner(System.
                                      in);
        System.out.println("Enter the
            coefficient a:");
        & double a = scanner.nextDouble();
        System.out.println("Enter the
                coefficient b:");;
        double b = scanner.nextDouble();
        System.out.println("Enter the
                coefficient c:");
        double c = scanner.nextDouble();
        double discriminant = b*b-4*a*c;
        if(discriminant>0)
        {
            double root1 =(-b+Math.sqrt(
                    discriminant))/(2*a);
            double root2 =(-b-Math.sqrt
```

```
    (discriminant))/(2*a);
System.out.println("The real solution
    - ons ar =");
System.out.println("x1 =" + root1);
System.out.println("x2 =" + root2);
}
else if(discriminant == 0)
{
    double root = -b/(2*a);
    System.out.println("There is
    one real solution :");
    System.out.println("x =" + root);
}
else
{
    System.out.println("There are no
    real solutions.");
}
scanner.close();
}
}
```

Output :-

```
Enter the coefficient a:
1
Enter the coefficient b:
3
Enter the coefficient bc:
1
The real solutions are:
x1 = -0.3819660112501051
x2 = -2.618033988749895
```

```
Enter co-efficient of x square
2
Enter co-efficient of x
4
Enter the constant
2
 The roots are equal -1.0
```

```
Enter co-efficient of x square
4
Enter co-efficient of x
4
Enter the constant
2
No real root exists
```

```
Enter co-efficient of x square
2
Enter co-efficient of x
3
Enter the constant
1
The roots are-0.5and-1.0
```

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.

```
import java.util.Scanner;

class Student {
    String name;
    String usn;
    int[] credits;
    int[] marks;
    int numberOfSubjects;

    void takeDetails() {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter your name: ");
        name = input.nextLine();
        System.out.print("Enter your USN: ");
        usn = input.nextLine();
        System.out.print("Enter number of subjects: ");
        numberOfSubjects = input.nextInt();

        credits = new int[numberOfSubjects];
        marks = new int[numberOfSubjects];

        for (int i = 0; i < numberOfSubjects; i++) {
            System.out.print("Enter the number of credits for subject " + (i + 1) + ": ");
            credits[i] = input.nextInt();
            System.out.print("Enter the marks for subject " + (i + 1) + ": ");
            marks[i] = input.nextInt();
        }
    }

    int gradePoints(int i) {
        if (marks[i] >= 90) return 10;
        else if (marks[i] >= 80) return 9;
        else if (marks[i] >= 70) return 8;
        else if (marks[i] >= 60) return 7;
        else if (marks[i] >= 50) return 6;
```

```
        else if (marks[i] >= 40) return 5;
        else return 0;
    }

    double calculateSGPA() {
        int totalGradePoints = 0;
        int totalCredits = 0;

        for (int i = 0; i < numberOfSubjects; i++) {
            totalCredits += credits[i];
            totalGradePoints += credits[i] * gradePoints(i);
        }

      return totalGradePoints/totalCredits;
    }
}

public class student_info {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
          System.out.print("Enter no of students ");
          int no=input.nextInt();
          for(int i = 0; i < no; i++)
          {

                Student student = new Student();
                student.takeDetails();
                System.out.println("Your total SGPA is: " + student.calculateSGPA());
          }

    }
}
import java.util.Scanner;

class Student {
    String name;
    String usn;
    int[] credits;
    int[] marks;
    int numberOfSubjects;
```

```
void takeDetails() {
    Scanner input = new Scanner(System.in);
    System.out.print("Enter your name: ");
    name = input.nextLine();
    System.out.print("Enter your USN: ");
    usn = input.nextLine();
    System.out.print("Enter number of subjects: ");
    numberOfSubjects = input.nextInt();

    credits = new int[numberOfSubjects];
    marks = new int[numberOfSubjects];

    for (int i = 0; i < numberOfSubjects; i++) {
        System.out.print("Enter the number of credits for subject " + (i + 1) + ": ");
        credits[i] = input.nextInt();
        System.out.print("Enter the marks for subject " + (i + 1) + ": ");
        marks[i] = input.nextInt();
    }
}

int gradePoints(int i) {
    if (marks[i] >= 90) return 10;
    else if (marks[i] >= 80) return 9;
    else if (marks[i] >= 70) return 8;
    else if (marks[i] >= 60) return 7;
    else if (marks[i] >= 50) return 6;
    else if (marks[i] >= 40) return 5;
    else return 0;
}

double calculateSGPA() {
    int totalGradePoints = 0;
    int totalCredits = 0;

    for (int i = 0; i < numberOfSubjects; i++) {
        totalCredits += credits[i];
        totalGradePoints += credits[i] * gradePoints(i);
    }
```

```
     return totalGradePoints/totalCredits;
   }
}

public class student_info {
   public static void main(String[] args) {
      Scanner input = new Scanner(System.in);
        System.out.print("Enter no of students ");
        int no=input.nextInt();
        for(int i = 0; i < no; i++)
        {

                Student student = new Student();
                student.takeDetails();
                System.out.println("Your total SGPA is: " + student.calculateSGPA());
        }

   }
}
```

```
Enter no of students 2
Enter your name: Chethan
Enter your USN: 074
Enter number of subjects: 3
Enter the number of credits for subject 1: 2
Enter the marks for subject 1: 92
Enter the number of credits for subject 2: 4
Enter the marks for subject 2: 88
Enter the number of credits for subject 3: 3
Enter the marks for subject 3: 80
Your total SGPA is: 9.0
Enter your name: raja
Enter your USN: 070
Enter number of subjects: 2
Enter the number of credits for subject 1: 3
Enter the marks for subject 1: 99
Enter the number of credits for subject 2: 1
Enter the marks for subject 2: 69
Your total SGPA is: 9.0
```

Date __/__/__
Page _____

Program-2
SGPA calculation

```
8.2. import java.util.Scanner;
class Student{
String usn;
String name;
int[] credits
int[] marks
int numSubjects;

Student(int noofsubjects)
{
    credits = new int[noofsubjects];
    marks = new int[noofsubjects];
}

public void acceptDetails()
{
  Scanner scanner = new Scanner(System
                                  in);

System.out.println("Enter your USN:");
USN = Scanner.nextLine();
System.out.println("Enter your Name:");
 name = scanner.nextline();
 for(int i=0; i< credits.length(); i++)
 {
    System.out.println("credits for
                        Subjects" + (i+1)+":");
    credit[i] = scanner.nextInt();
    System.out.println("Marks for Subject
                        (i+1) + ":");
    marks[i] = scanner.nextInt();
 }
}
```

```
public void DisplayDetails()
{
  System.out.println("\n Student Details:
    \n USN:" + usn + "\n Name:" + name);
  for(int i = 0; i < credits.length; i++)
  {
    System.out.println(" Subject" + (i+1)
      + ":" + "Credits =" + credits[i] +"
      ", Marks =" + marks[i]);
  }
}
public double calculateSGPA()
{
  double totalpoints = 0;
  int totalcredits = 0;
  for(int i = 0; i < credits.length; i++)
  {
    totalpoints += (marks[i] * credits[i]);
    total credits += credits[i];
  }
  return totalcredits = =
  double SGPA;
  SGPA = totalpoints / total.credits;
System.out.println("SGPA =" + SGPA);
}
public static void main(String[]
                         args)
{ Scanner scanner = new Scanner
                    (System.in);
  System.out.println("Enter the
    no. of students =");
  Student student = new Student
      (Scanner.nextInt());
```

Date __/__/__
Page _____

```
Student.acceptDetails();
Student.displayDetails();
Syst.out.pri
Student.calculateSGPA();
}
}
```

Output -

```
Enter the Number of Subjects: 2
Enter your USN: 1BM23CS275
Enter your Name: Jayanth
Credits for Subject 1: 2
Marks for Subject 1: 90
Credits for Subject 2: 4
Marks for Subject 2: 95

** Student Details **
USN: 1BM23CS275
Name: Jayanth
Subject 1: Credits=2 Marks=90
Subject 2: Credits=4 Marks=95
SGPA = 9.5
```

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.

```
import java.util.Scanner;
public class Book
{
	private String name;
	private String author;
	private double price;
	private int num_pages;

public Book(String name, String author, double price, int num_pages)
{
	this.name=name;
	this.author=author;
	this.price=price;
	this.num_pages=num_pages;
}
public String getName()
{
	return name;
}
public void setName(String name)
{
	this.name=name;
}
public String getAuthor()
{
	return author;
}
public void setAuthor(String author)
{
	this.author=author;
}
public double getPrice()
{
```

```
        return price;
}
public void setPrice(double price)
{
        this.price=price;
}
public int getNumPages()
{
        return num_pages;
}
public void setNumPages(int num_pages)
{
        this.num_pages=num_pages;
}
@Override
public String toString()
{
        return "\nName:" + name + "\nAuthor:" + author + "\nPrice:" + price +
"\nNumber of pages:" + num_pages;
}
public static void main(String args[])
{
        Scanner input = new Scanner(System.in);
        System.out.print("\n Enter the number of books:");
        int n = input.nextInt();
        input.nextLine();
        Book[] books =new Book[n];
        for(int i=0;i<n;i++)
        {
                System.out.print("\n Enter the name of the book:");
                String name=input.nextLine();
                System.out.print("\n Enter the author name:");
                String author = input.nextLine();
                System.out.print("\n Enter the price of the book:");
                double price = input.nextDouble();
                System.out.print("\n Enter the number of pages:");
                int num_pages = input.nextInt();
                input.nextLine();
                books[i] = new Book (name,author,price,num_pages);
        }
```

```
        for (Book book:books)
        {
        System.out.println(book.toString());
        }

input.close();
}
}
```

```
 Enter the number of books:2

 Enter the name of the book:Harry Potter

 Enter the author name:J K Rowling

 Enter the price of the book:199.99

 Enter the number of pages:155

 Enter the name of the book:The wings of fire

 Enter the author name:APJ Abdul Kalam

 Enter the price of the book:99

 Enter the number of pages:150

Name:Harry Potter
Author:J K Rowling
Price:199.99
Number of pages:155

Name:The wings of fire
Author:APJ Abdul Kalam
Price:99.0
Number of pages:150
```

Date __/__/__
Page _____

Program - 3
Book Details

```
import java.util.Scanner;
public class Book {
    private String name;
    private String author;
    private double price;
    private int num_pages;
    Book(String name, String auth
        double price, int num_page
    {
        this.name = name;
        this.author = author;
        this.price = price;
        this.num_pages = num_pages;
    }

public String getName()
{
    return name;
}

public void SetName(String name)
{
    this.name = name;
}
public String getAuthor()
{
    return author;
}
public void SetAuthor(String
                author)
```

```
    {
        this.author = author;
    }
public double getPrice()
{
    return price;
}
public void setPrice(double price)
{
    this.price = price;
}
public int getnum_pages()
{
    return num_pages;
}
public void setnum_pages(int
                         num_pages
{
    this.num_pages = num_pages
}

public String toString()
{
    return "*** Book Details ***\n
        + "Name of the book = "+ name+"\n
        + "Book Author = " + Author + "\n"+
        + "Book Price" + price + "\n" +
        "Number of Pages" + num_pages+
                                "\n";
    }
}
```

Date __/__/____
Page _____

```
public class Main
{
  public static void main(String[] args)
{
  Scanner scanner = new Scanner
                    (System.in);
  System.out.println("Enter the
  number of books :");
  int n = scanner.nextInt();
  Book[] books = new Books[n];
  for(int i=0; i<n; i++)
{
  System.out.println("Enter the details
    of" +(i+1) + ":");
  System.out.println("Enter book
                    name =:");
    String name = scanner.nextLine();
System.out.println("Enter Author
                    name =:");
  String author = scanner.nextLine();
System.out.println("Enter Price =");
  double price = scanner.nextDouble();
System.out.println("Enter the no.
                of Pages =");
  int num_pages = scanner.nextInt();
  books[i] = new Book('name, author,
                price, num_pages);
}
System.out.println("Book Details");
for(int i=0; i<n; i++)
{
```

Date __/__/__
Page _____

```
        System.out.println(book1[i].toStri-
                                    -ng());
    }
}
}
```

Output -

Enter the number of books : 2
Enter the details of book 1 :

Enter the book name : Java
Enter the book author : DK
Enter the book price : 200
Enter no. of pages = 205

Enter the details of book 2 :
Enter the book name : Java 2
Enter the book author : MK
Enter the book price : 250
Enter the book pages : 100

* * * Book Details * * *
Book Name : Java
Author Name : DK
Price : 200
Pages : 205
* * * Book Details * * *
Book name : Java 2
Author name : MK
Price : 250
Pages : 100

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.

```
abstract class Shape{
	int x,y;
	abstract void printarea();
}

class Rectangle extends Shape{
	Rectangle (int l, int b){
		x=l;
		y=b;
	}

@Override
void printarea(){
	int area = x+y;
	System.out.println("Area of rectangle is " + area);
	}
}

class Triangle extends Shape{
	Triangle (int b, int h){
		x=b;
		y=h;
	}

@Override
void printarea(){
	double area = 0.5*x*y;
	System.out.println("Area of triangle is " + area);
	}
}

class Circle extends Shape{
```

```
        int radius;
        Circle (int radius){
                this.radius=radius;
        }

@Override
void printarea(){
        double area = 3.14*radius*radius;
        System.out.println("Area of circle is " + area);
        }
}

public class Geometry{
        public static void main (String args[]){
                Shape r=new Rectangle(5,11);
                Shape t=new Triangle(20,9);
                Shape c=new Circle (5);
                r.printarea();
                t.printarea();
                c.printarea();
            System.out.print("Chethan K S\n1BM23CS074");
        }
}
```

```
Area of rectangle is 16
Area of triangle is 90.0
Area of circle is 78.5
```

Date __/__/__
Page

Program-4
Shape Area

Q.4. Develop a java program to create an abstract class name Shape that contains two integers and an empty method name printArea(). Prone 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.

A.4.
```
abstract class Shape
{
    int dim1;
    int dim2;
    public Shape(int dim1, int dim2)
    {
        this.dim1 = dim1;
        this.dim2 = dim2;
    }
    public void abstract printArea();
}
class Rectangle extends Shape
{
    public Rectangle(int length, int breadth)
    {
        super(length, breadth);
    }
```

```
public void printArea()
{
    int area = dim1 * dim2;
    System.out.println("Area of
    Rectangle:" + area);
}
}
class Triangle extends Shape
{
    public Triangle(int base, int
                    height)
    {
        super(base, height);
    }
public void printArea()
{
    int double area = 0.5 * dim1 * dim2;
    System.out.println("Area of
    Triangle:" + area);
}
}

class Circle extends Shape
{
    public Circle(int radius)
    {
        super(radius, 0);
    }
public void printArea()
{
double int area = 3.14 * dim1 * dim1;
System.out.println("Area of Circle="
                    + area);
```

```
    }
}
class Main
{
  public static void main(String[]
                                args)
{
  Shape R = new Rectangle(10,20);
  Shape T = new Triangle(10,30);
  Shape C = new Circle(10);
  R.printArea();
  T.printArea();
  C.printArea();
  }
}
```

Output -

```
Area of Rectangle: 200.0
Area of Triangle: 150.0
Area of Circle: 314.0
```

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.

```
import java.util.*;

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

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

   void deposit(double amount) {
       balance += amount;
```

```
        System.out.println("Deposit successful. New balance: " + balance);
    }

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

    abstract void computeInterest();

    abstract void withdraw(double amount);
}

class SavAcct extends Account {
    final double interestRate = 0.04;

    SavAcct(String customerName, int accountNumber, double balance) {
        super(customerName, accountNumber, "Savings", balance);
    }

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

    @Override
    void withdraw(double amount) {
        if (balance >= amount) {
            balance -= amount;
            System.out.println("Withdrawal successful. New balance: " + balance);
        } else {
            System.out.println("Insufficient balance.");
        }
    }
}

class CurAcct extends Account {
    double minBalance = 1000.00;
    double charge = 50.00;
```

```
    double[] chequeTransactions = new double[100];
    int chequeId = 0;

    CurAcct(String customerName, int accountNumber, double balance) {
        super(customerName, accountNumber, "Current", balance);
    }

    @Override
    void computeInterest() {
        System.out.println("Interest cannot be calculated for a Current Account.");
    }

    @Override
    void withdraw(double amount) {
        if (balance >= amount) {
            balance -= amount;
            if (balance >= minBalance) {
                System.out.println("The updated balance is: " + balance);
            } else {
                balance -= charge;
                System.out.println("Penalty of 50.0 has been deducted. The new balance is: "
+ balance);
            }
            chequeTransactions[chequeId] = amount;
            chequeId++;
        } else {
                System.out.println("Insufficient balance. The withdrawal amount is greater
than balance.");
        }
    }

    void displayTransactions() {
        for (int i = 0; i < chequeId; i++) {
            System.out.println("Transaction " + (i + 1) + ": " + chequeTransactions[i]);
        }
    }
}

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

```
Scanner input = new Scanner(System.in);

System.out.println("Enter account type:");
System.out.println("1. Savings");
System.out.println("2. Current");
int choice = input.nextInt();
input.nextLine();

System.out.println("Enter your name:");
String name = input.nextLine();

System.out.println("Enter your account number:");
int accountNumber = input.nextInt();

System.out.println("Enter the initial balance:");
double balance = input.nextDouble();

Account account;

if (choice == 1) {
   account = new SavAcct(name, accountNumber, balance);
} else {
   account = new CurAcct(name, accountNumber, balance);
}

int exit = 0;

while (exit != 1) {
   System.out.println("\nEnter the function to be done:");
   System.out.println("1. Deposit");
   System.out.println("2. Display balance");
   System.out.println("3. Compute and deposit interest");
   System.out.println("4. Withdrawal");
   System.out.println("5. Exit");

   int func = input.nextInt();

   switch (func) {
      case 1:
         System.out.println("Enter deposit amount:");
```

```
                double depAmount = input.nextDouble();
                account.deposit(depAmount);
                break;

            case 2:
                account.displayBalance();
                break;

            case 3:
                account.computeInterest();
                break;

            case 4:
                System.out.println("Enter withdrawal amount:");
                double withdrawAmount = input.nextDouble();
                account.withdraw(withdrawAmount);
                break;

            case 5:
                exit = 1;
                System.out.println("Exiting");
                break;

            default:
                System.out.println("Invalid input");
        }

        if (choice == 2) {
            ((CurAcct) account).displayTransactions();
        }
    }
    System.out.print("Chethan K S\n1BM23CS074");
    input.close();
  }
}
```

```
Enter account type:
1. Savings
2. Current
1
Enter your name:
Chethan
Enter your account number:
22
Enter the initial balance:
3333

Enter the function to be done:
1. Deposit
2. Display balance
3. Compute and deposit interest
4. Withdrawal
5. Exit
1
Enter deposit amount:
555
Deposit successful. New balance: 3888.0

Enter the function to be done:
1. Deposit
2. Display balance
3. Compute and deposit interest
4. Withdrawal
5. Exit
2
Balance: 3888.0

Enter the function to be done:
1. Deposit
2. Display balance
3. Compute and deposit interest
4. Withdrawal
5. Exit
3
Interest added. New balance: 4043.52

Enter the function to be done:
1. Deposit
2. Display balance
3. Compute and deposit interest
4. Withdrawal
5. Exit
4
Enter withdrawal amount:
566
Withdrawal successful. New balance: 3477.52

Enter the function to be done:
1. Deposit
2. Display balance
3. Compute and deposit interest
4. Withdrawal
5. Exit
5
Exiting
```

```
Enter account type:
1. Savings
2. Current
2
Enter your name:
Chethan
Enter your account number:
2645
Enter the initial balance:
2534

Enter the function to be done:
1. Deposit
2. Display balance
3. Compute and deposit interest
4. Withdrawal
5. Exit
1
Enter deposit amount:
3145
Deposit successful. New balance: 5679.0

Enter the function to be done:
1. Deposit
2. Display balance
3. Compute and deposit interest
4. Withdrawal
5. Exit
2
Balance: 5679.0

Enter the function to be done:
1. Deposit
2. Display balance
3. Compute and deposit interest
4. Withdrawal
5. Exit
3
Interest cannot be calculated for a Current Account.

Enter the function to be done:
1. Deposit
2. Display balance
3. Compute and deposit interest
4. Withdrawal
5. Exit
4
Enter withdrawal amount:
5131655
Insufficient balance. The withdrawal amount is greater than balance.

Enter the function to be done:
1. Deposit
2. Display balance
3. Compute and deposit interest
4. Withdrawal
5. Exit
5
Exiting
```

update the balance.

```
import java.util.Scanner;
class Account
{
  String name;
  int accno;
  double balance;
  String accounttype
  public Account(String name, int
    accno, double balance, String
  {                          accountype
    this.name = name;
    this.accno = accno;
    this.balance = balance;
  } this.accounttype = accountype
  }
  public void displayaccountDetails(
{
  System.out.println("Account Holder
                          name);
  System.out.println("Account number:
                          accno);
  System.out.println("Account Type:"
                          +accountype);
  System.out.println("Balance :"+balan

  public void deposit(double amount
{
        balance += amount;
  System.out.println("Deposited:"+amount);
  System.out.println("New Balance:"+balance);
```

Date __/__/__
Page _____

```
public void withdraw(double amount)
{
    balance -= amount;
}
}

class SavAcct extends Account
{
  private final double interestrate
                          = 0.04;
  public SavAcct(String customername,
String accountnumber, double balance)
{
    super(customername, accountNumber,
          "Savings", balance);
public void computeAndDepositInterest
()
{
  double interest = balance * interestRate;
  balance += interest;
  System.out.println("Interest Deposi-
-ted : " + interest);
System.out.println("New balance
after Interest :" + balance);

public void withdraw(double amount)
{
   if (balance >= amount)
   {
        balance -= amount;
      System.out.println("withdrawn:"
System.out.println("New " + amount);
                Balance:" + balance);
   }
```

```
else
{
System.out.println("Insufficient balance to withdraw" + amount);
}
}
}

class CurAcct extend Account
{
private final double minimum balance = 1000.0;
private final double penalty = 50.0;
pubic CurAcct(String customername, accountNumber, "current", balance);

public void withdraw(double amount)
{
if(balance - amount >= minimum balance)
{
balance -= amount;
System.out.println("withdrawn:" + amount);
System.out.println("New balance:" + balance);
}
else
{
System.out.println("Balance is below minimum required. Penalty imposed");
balance -= penalty;
```

```
CurAcct currentAccount = new
    curAcct("Jane Smith", "CA987
                          "1500.0);
    currentaccount.displayAccountDetai
currentaccount.deposit(500.0);
currentaccount.withdraw(1200.0);
currentaccount.checkminimumbalance
currentAccount.displayAccountDetails
  }
}
```

Output -

Account Holder: JohnDoe
Account Number: SA12345
Account type: Savings
Balance: 5000.0
Deposited: 20000
New Balance: 7000.0
Interest Deposited: 280.0
New balance after Interest: 7280.0
Withdrawn: 1000.0
New Balance: 6280.0

Account Holder: Jane Smith
Account Number: CA98765
Account Type: Current
Balance: 1500.0
Deposited: 500.0
New Balance: 2000.0
Withdrawn: 1200.0
New Balance: 800.0
Servie charge imposed due to low balance.

Date __/__/__
Page _____

New balance after Service charge: 750.0
Account Holder: Jane Smith
Account Number: CA98765
Account type: Current
Balance: 750.0

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 Student {
   public int[] internalMarks = new int[5];

   public void setInternalMarks(int[] marks) {
      for (int i = 0; i < 5; i++) {
         internalMarks[i] = marks[i];
      }
   }

   public int[] getInternalMarks() {
      return internalMarks;
   }
}
package CIE;

public class Student {
   public String usn;
   public String name;
   public int sem;
}
package SEE;

import CIE.Student;

public class External extends Student {
   public int[] seeMarks = new int[5];
```

```
    public void setSEEMarks(int[] marks) {
        for (int i = 0; i < 5; i++) {
            seeMarks[i] = marks[i];
        }
    }

    public int[] getSEEMarks() {
        return seeMarks;
    }
}
import CIE.*;
import SEE.*;

import java.util.Scanner;

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

        Student[] students = new Student[n];
        Internals[] internalMarks = new Internals[n];
        External[] seeMarks = new External[n];

        for (int i = 0; i < n; i++) {
            students[i] = new Student();
            internalMarks[i] = new Internals();
            seeMarks[i] = new External();

            System.out.print("Enter USN for Student " + (i + 1) + ": ");
            students[i].usn = sc.next();
            sc.nextLine();

            System.out.print("Enter Name for Student " + (i + 1) + ": ");
            students[i].name = sc.nextLine();

            System.out.print("Enter Semester for Student " + (i + 1) + ": ");
            students[i].sem = sc.nextInt();
```

```
            int[] internals = new int[5];
            System.out.println("Enter Internal Marks (5 courses) for Student " + (i + 1) + ":
");
            for (int j = 0; j < 5; j++) {
                internals[j] = sc.nextInt();
            }
            internalMarks[i].setInternalMarks(internals);

            int[] see = new int[5];
            System.out.println("Enter SEE Marks (5 courses) for Student " + (i + 1) + ": ");
            for (int j = 0; j < 5; j++) {
                see[j] = sc.nextInt();
            }
            seeMarks[i].setSEEMarks(see);
        }

        System.out.println("\nFinal Marks of Students:");
        for (int i = 0; i < n; i++) {
            System.out.println("\nStudent " + (i + 1) + ": " + students[i].name + " (USN: "
+ students[i].usn + ")");
            System.out.println("Course\tInternal\tSEE\tFinal Marks");
            for (int j = 0; j < 5; j++) {
                int finalMark = internalMarks[i].getInternalMarks()[j] +
seeMarks[i].getSEEMarks()[j];
                System.out.println("Course " + (j + 1) + ":\t" +
internalMarks[i].getInternalMarks()[j] + "\t\t" + seeMarks[i].getSEEMarks()[j] + "\t" +
finalMark);
            }
        }

        sc.close();
    }
}
```

```
Enter the number of students: 1
Enter USN for Student 1: 1BM23CS074
Enter Name for Student 1: CHETHAN K S
Enter Semester for Student 1: 3
Enter Internal Marks (5 courses) for Student 1:
40
39
36
35
31
Enter SEE Marks (5 courses) for Student 1:
50
49
47
45
49

Final Marks of Students:

Student 1: CHETHAN K S (USN: 1BM23CS074)
Course  Internal        SEE     Final Marks
Course 1:       40              50      90
Course 2:       39              49      88
Course 3:       36              47      83
Course 4:       35              45      80
Course 5:       31              49      80
```

Date / /
Page

Program - 6
Package CIE

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 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 two packages in a file that declares the final marks of n students in all five courses.

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

```
public void display()
{
  System.out.println("Name:" + name);
  System.out.println("USN:" + usn);
  System.out.println("Sem:" + sem);

class Internals extends Student
{
   int[] Internalmarks = new Int[5];
   public Internals(String name, int usn,
                    int sem, int[] Internalmarks
   {
     super(name, usn, sem);
     this.Internalmarks = internalmarks;
   }
public void display()
{
   for(i=0; i<5; i++)
   {
     System.out.println("Course"+(i+1)+":"
                        + internalmarks[i]);
   }
}
```

SEE package External.java

```
package SEE
import CIE.Student;
public class External extends Student{
   int[] externalm = new externalm[5];
   public External(String name, int usn,
                   int sem, int[] external
     super(name, usn, sem);
```

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

Main.Java

```
import CIE.Student;
import SEE.Externals;
import java.util.Scanner;
public class Main{
public static void main(String args
{
    Scanner sc=new Scanner(System.i
            sc.nextline();

System.out.println("Enter the
Number of Students:");
    int n=sc.nextInt();
Student[] Student=new Student[n]
Externals[] External=new Externalm
```

```
for(int i=0; i<n; i++){
System.out.println("Name:");
String name = sc.nextLine();
System.out.println("USN:");
int usn = sc.nextInt();
Student[i] = new Student(name, usn,
                         sem);
}
for(int i=0; i<5; i++)
{
System.out.println("Enter Internal
marks of Sub"+(i+1)"+");
  Internal[i] = sc.nextInt();
System.out.println("Enter External
marks of Sub"+(i+1)+":");
  External[i] = sc.nextInt();
}
}
Student[i] = new Student(name, usn,
  sem, internalm);
External[i] = new External
for(int i=0; i<n; i++)
{
System.out.println("Student"+(i+1)"
                   details:");
  Student[i].display();
for(int i=0; i<5; i++)
{
System.out.println("Internal mar
of Subjects" + Internal[i].displa
```

```
System.out.println("\n Final
marks of Students :");
for(int i=0;i<n;i++)
{
  System.out.println("\n Student "
+(i+1)+":"+Students[i].name+
  "(USN:"+Students[i].USN+")");

System.out.println("Course | Interna
l\t SEE\t Final Marks.");
for(int j=0;j<5;j++)
{
 int finalmark=internalMarks[i].
 getInternalMarks()[j].getInternal
 Marks()[j].getSEEMarks()[j];

System.out.println("Course"+
(j+1)+":\t"+internalMarks[i].
getInternalMarks()[j]+"\t\t"+
SEEMarks[i].getSEEMarks()[j]+
"\t"+finalmarks);
}
}
System.out.print("Jayanta
Bhathan K gV
\n1BM23IS275");
sc.close();
}
}
```

Date __/__/__
Page _____

Output -

Enter the Number of Students: 1
Enter USN for student 1: 1BM23IS275
Enter Name for student 1: Jayanth
Enter Semester for student 1: 3
Enter Internal Marks (5 courses) for student 1:
40
39
36
35
31
Enter SEE Marks (5 courses) for student 1:
50
49
47
45
49
Final Marks of Students:
Student 1: ~~Brothro~~ Jayanth (USN: 1BM23IS275)

Course	Internal	SEE	Final Marks
Course 1:	40	50	90
Course 2:	39	49	88
Course 3:	36	47	83
Course 4:	35	45	80
Course 5:	31	49	80

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 uses both father and son's age and throws an exception if son's age is >=father's age.

```
class WrongAge extends Exception {
   public WrongAge(String message) {
      super(message);
   }
}

class Father {
   int fatherAge;

   public Father(int fatherAge) throws WrongAge {
      if (fatherAge < 0) {
         throw new WrongAge("Father's age cannot be negative: " + fatherAge);
      }
      this.fatherAge = fatherAge;
      System.out.println("Father's age set to " + this.fatherAge);
   }
}

class Son extends Father {
   int sonAge;

   public Son(int fatherAge, int sonAge) throws WrongAge {
      super(fatherAge);

      if (sonAge < 0) {
         throw new WrongAge("Son's age cannot be negative: " + sonAge);
      }

      if (sonAge > fatherAge) {
            throw new WrongAge("Son's age cannot be greater than father's age: " +
sonAge);
```

```
        }

        this.sonAge = sonAge;
        System.out.println("Son's age set to " + this.sonAge);
    }
}

public class labexception {
    public static void main(String[] args) {
        try {
            System.out.println("Test case 1:");
            int father1 = 40;
            int son1 = 15;
            int father2 = 40;
            int son2 = -5;

            try {
                Son s1 = new Son(father1, son1);
            } catch (WrongAge e) {
                System.out.println("Error: " + e.getMessage());
            }

            try {
                Son s2 = new Son(father2, son2);
            } catch (WrongAge e) {
                System.out.println("Error: " + e.getMessage());
            }

            System.out.println("\nTest case 2:");
            int father3 = -30;
            int son3 = 10;

            try {
                Son s3 = new Son(father3, son3);
            } catch (WrongAge e) {
                System.out.println("Error: " + e.getMessage());
            }

            System.out.println("\nTest case 3:");
            int father4 = 40;
```

```
            int son4 = 50;

            try {
                Son s4 = new Son(father4, son4);
            } catch (WrongAge e) {
                System.out.println("Error: " + e.getMessage());
            }

        } finally {
            System.out.print("Jayanth\n1BM23IS275");
        }
    }
}
```

```
Test case 1:
Father's age set to 40
Son's age set to 15
Father's age set to 40
Error: Son's age cannot be negative: -5

Test case 2:
Error: Father's age cannot be negative: -30

Test case 3:
Father's age set to 40
Error: Son's age cannot be greater than father's age: 50
```

Date __/__/__
Page _____

Program-7
Exception Handling

Q) Write a program that demonstrate 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 uses both father and Sons age and throws an exception if Sons age is >= fathers age.

A)
```
class WrongAge extends Exception
{

    WrongAge(String message)
    {
        super(message);
    }
}

class Father
{
    int fatherage;
    Father(int fatherage) throws
                        WrongAge
    {
```

```
if(fatherage<0)
{
  throw new WrongAge("Fathers
  age cannot be negative:"+
                    fatherage);
}
  this.fatherage = fatherage;
System.out.println("Fathers age
  set to" + this.fatherage);
}
}

class Son extends Father
{
  int sonAge;
  Son(int fatherage, int sonage)
                    throws WrongAge
{
    super(fatherage);
  if(sonAge<0)
{
throw new WrongAge("Sons
  age cannot be negative:"+sonage
}
if(sonAge > fatherage)
{
  throw new WrongAge("Sons
  age cannot be greater than
  fathers age:" + sonage);
}
  this.sonage = sonage;
System.out.println("Sons age
  set to" + this.sonAge);
```

```
}
}
public class labexception
{
  public static void main(String args)
  {
    System.out.println("Test case1:");
    int father1=40;
    int son1=15;
    int father2=40;
    int son2=-5;

    try
    {
      Son s1 = new Son(father1,son1);
    }
    catch(WrongAge e)
    {
      System.out.println("Error:"
                         + e.getMessage());
    }

    try
    {
      Son s2 = new Son(father2,son2);
    }
    catch(WrongAge e)
    {
      System.out.println("Error:"+
                         e.getMessage());
    }
```

```
System.out.println("\n Test case 2:");
int father3 = -30;
int son3 = 10;

try{
  son s3 = new son(father3, son3);
  }
catch(WrongAge e)
{
  System.out.println("Error:" +
                     e.getMessage());
}
System.out.println("\n Test case 3:");
int father4 = 40;
int son4 = 50;

try{
  son s4 = new son(father4, son4);
  }
catch(WrongAge e)
{
System.out.println("Error:"
                   + e.getMessage());
}
} finally {
    System.out.println("@Jayanth
                       IBM23IS275");

  }
}
}
```

Date __/__/__
Page _____

Output -

Test case 1:
Fathers age set to 40
Sons age set to 45
Fathers age set to 40.
Error: Sons age cannot be negative

Test case 2:
Error: Fathers age cannot be negative

Test case 3:
Fathers age set to 40.

Error: Sons age cannot be grea
than fathers age: 50

Jayanth
1BM23IS275

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 CollegeThread extends Thread {
   public void run() {
      try {
         while (true) {
            System.out.println("BMS College of Engineering");
            Thread.sleep(10000); // 10 seconds
         }
      } catch (InterruptedException e) {
         System.out.println("Thread Interrupted: " + e.getMessage());
      }
   }
}

class CSEThread extends Thread {
   public void run() {
      try {
         while (true) {
            System.out.println("CSE");
            Thread.sleep(2000); // 2 seconds
         }
      } catch (InterruptedException e) {
         System.out.println("Thread Interrupted: " + e.getMessage());
      }
   }
}

public class DisplayThreads {
   public static void main(String[] args) {
      // Creating threads
      CollegeThread t1 = new CollegeThread();
      CSEThread t2 = new CSEThread();

      // Starting threads
      t1.start();
      t2.start();
```

```
    }
}
```

```
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
```

Date __/__/__
Page _____

Program - 8

Q) Write a program which creates two threads, one thread displaying "BMS college of Engineering" once every 10 seconds and another displaying "CSE" once every two seconds.

A)

```
class CollegeThread extends Thread
{
  public void run()
  {
    try{
      while(true)
      {
      System.out.println("BMS
      College of Engineering");
      Thread.sleep(10000);
      }
    catch(InterruptedException e)
    {
      System.out.println("Thread
      Interrupted:" + e.getMessage
    }
  }
}
```

```
class CSEThread extends Thread
{
  public void run()
  {
    try{
      while(true)
      {
        System.out.println("CSE");
        Thread.sleep(2000);
      } }
    catch(InterruptedException e)
    {
      System.out.println("Thread
          Interrupted:"+e.get
          Message());
    }
  }
}

public class DisplayThreads
{
  public static void main(Strin
                          [] args)
  {
    CollegeThread t1 = new CollegeTh
    CSEThread t2 = new CSEThread(
    t1.start();
    t2.start();
  }
}
```

Date __/__/__
Page _____

Output -

CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
BMS College of Engineering
....

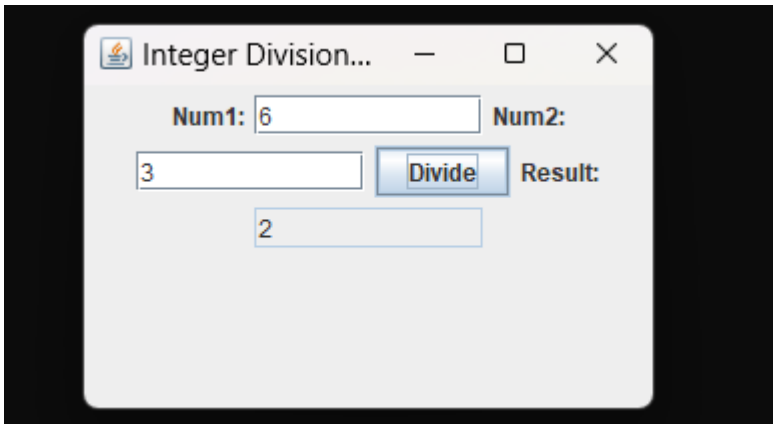
4/7/12

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.

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

public class DivisionApp extends JFrame {
    private JTextField num1Field, num2Field, resultField;
    private JButton divideButton;

    public DivisionApp() {
        setTitle("Integer Division App");
        setLayout(new FlowLayout());
        setSize(300, 200);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JLabel num1Label = new JLabel("Num1:");
        num1Field = new JTextField(10);

        JLabel num2Label = new JLabel("Num2:");
        num2Field = new JTextField(10);

        JLabel resultLabel = new JLabel("Result:");
        resultField = new JTextField(10);
        resultField.setEditable(false);

        divideButton = new JButton("Divide");

        add(num1Label);
        add(num1Field);
        add(num2Label);
        add(num2Field);
```

```
        add(divideButton);
        add(resultLabel);
        add(resultField);

        divideButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                try {
                    int num1 = Integer.parseInt(num1Field.getText());
                    int num2 = Integer.parseInt(num2Field.getText());

                    int result = num1 / num2;
                    resultField.setText(String.valueOf(result));

                } catch (NumberFormatException ex) {
                        JOptionPane.showMessageDialog(DivisionApp.this, "Please enter valid
integers.", "Input Error", JOptionPane.ERROR_MESSAGE);
                } catch (ArithmeticException ex) {
                    JOptionPane.showMessageDialog(DivisionApp.this, "Division by zero is
not allowed.", "Arithmetic Error", JOptionPane.ERROR_MESSAGE);
                } finally {


                }
            }
        });
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            @Override
            public void run() {
                new DivisionApp().setVisible(true);
            }
        });
    }
}
```


Integer Division...
Num1:
6
Num2:
3
Divide
Result:
2