

LAB PROGRAM 1

```
Abstract.java          Quad.java          +  
import java.util.Scanner;  
  
class Quad {  
    public static void main(String[] args) {  
  
        double a, b, c, discriminant, root1, root2;  
        Scanner s1 = new Scanner(System.in);  
        System.out.print("Enter the coefficient a: ");  
        a = s1.nextDouble();  
  
        System.out.print("Enter the coefficient b: ");  
        b = s1.nextDouble();  
  
        System.out.print("Enter the coefficient c: ");  
        c = s1.nextDouble();  
  
        if (a==0 || b==0 ||c==0){  
            System.out.print("Invalid Input");  
        }  
        else{  
            discriminant = b * b - 4 * a * c;  
  
            if (discriminant >= 0) {  
  
                root1 = (-b + Math.sqrt(discriminant)) / (2 * a);  
                root2 = (-b - Math.sqrt(discriminant)) / (2 * a);  
  
                System.out.println("Real solutions:");  
                System.out.println("Root 1: " + root1);  
                System.out.println("Root 2: " + root2);  
            } else {  
  
                System.out.println("No real solutions, discriminant is  
negative.");  
            }  
        }  
    }  
}
```

```
(base) madhupandey@Madhus-MacBook-Air javalab % javav Quad.java  
zsh: command not found: javav  
(base) madhupandey@Madhus-MacBook-Air javalab % javac Quad.java  
(base) madhupandey@Madhus-MacBook-Air javalab % java Quad  
Enter the coefficient a: 1  
Enter the coefficient b: 5  
Enter the coefficient c: 2  
Real solutions:  
Root 1: -0.4384471871911697  
Root 2: -4.561552812808831  
(base) madhupandey@Madhus-MacBook-Air javalab %
```

QUE 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.

LAB PROGRAM 2

```
import java.util.Scanner;

class Students {
    String usn, name;
    int[] credits;
    int[] marks;
    Scanner s = new Scanner(System.in);

    Students(String usn, String name, int numSubjects) {
        this.usn = usn;
        this.name = name;
        this.credits = new int[numSubjects];
        this.marks = new int[numSubjects];
    }

    void acceptDetails() {
        System.out.println("Enter USN:");
        usn = s.nextLine();

        System.out.println("Enter Name:");
        name = s.nextLine();

        for (int i = 0; i < credits.length; i++) {
            System.out.println("Enter credits for subject " + (i + 1) + ":");
            credits[i] = s.nextInt();
        }

        for (int i = 0; i < marks.length; i++) {
            System.out.println("Enter marks for subject " + (i + 1) + ":");
            marks[i] = s.nextInt();
        }
    }

    void displayDetails() {
        System.out.println("Details of student:");
        System.out.println("USN = " + usn);
        System.out.println("Name = " + name);

        System.out.println("Credits of subjects:");
        for (int i = 0; i < credits.length; i++) {
            System.out.println("Subject " + (i + 1) + ": " + credits[i]);
        }

        System.out.println("Marks of subjects:");
        for (int i = 0; i < marks.length; i++) {
            System.out.println("Subject " + (i + 1) + ": " + marks[i]);
        }
    }

    double calSGPA() {
```

QUE 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.

```
(base) madhupandey@Madhus-MacBook-Air javalab % javac SGPA.java
(base) madhupandey@Madhus-MacBook-Air javalab % java SGPA
Enter the number of subjects:
3
Enter USN:
1BM22CS140
Enter Name:
Madhu
Enter credits for subject 1:
4
Enter credits for subject 2:
3
Enter credits for subject 3:
1
Enter marks for subject 1:
90
Enter marks for subject 2:
98
Enter marks for subject 3:
97
Details of student:
USN = 1BM22CS140
Name = Madhu
Credits of subjects:
Subject 1: 4
Subject 2: 3
Subject 3: 1
Marks of subjects:
Subject 1: 90
Subject 2: 98
Subject 3: 97
SGPA: 10.0
(base) madhupandey@Madhus-MacBook-Air javalab %
```

```

        for (int i = 0; i < marks.length; i++) {
            System.out.println("Subject " + (i + 1) + ": " + marks[i]);
        }

    double calSGPA() {
        double totalCredit = 0.0, totalGradePoints = 0.0;
        for (int i = 0; i < credits.length; i++) {
            totalCredit += credits[i];
            totalGradePoints += calGradePoints(marks[i]) * credits[i];
        }
        return totalGradePoints / totalCredit;
    }

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

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

            System.out.println("Enter the number of subjects:");
            int numSubjects = scanner.nextInt();

            Students s = new Students("1BM18CS001", "Riya", numSubjects);

            s.acceptDetails();
            s.displayDetails();

            double sgpa = s.calSGPA();
            System.out.println("SGPA: " + sgpa);
        }
    }
}

```

```

method PrintStream.println(char[]) is not applicable
    (actual and formal argument lists differ in length)
method PrintStream.println(String) is not applicable
    (actual and formal argument lists differ in length)
method PrintStream.println(Object) is not applicable
    (actual and formal argument lists differ in length)
[1 error]
(base) madhupandey@Madhus-MacBook-Air javalab % javac SGPA.java
(base) madhupandey@Madhus-MacBook-Air javalab % java SGPA
Enter the number of subjects:
3
Enter USN:
1BM22CS140
Enter Name:
Madhu
Enter credits for subject 1:
4
Enter credits for subject 2:
3
Enter credits for subject 3:
1
Enter marks for subject 1:
90
Enter marks for subject 2:
98
Enter marks for subject 3:
97
Details of student:
USN = 1BM22CS140
Name = Madhu
Credits of subjects:
Subject 1: 4
Subject 2: 3
Subject 3: 1
Marks of subjects:
Subject 1: 90
Subject 2: 98
Subject 3: 97
SGPA: 10.0
(base) madhupandey@Madhus-MacBook-Air javalab %

```

LAB PROGRAM 3

```
import java.util.Scanner;

class Students {
    String usn, name;
    int[] credits;
    int[] marks;
    Scanner s = new Scanner(System.in);

    Students(String usn, String name, int numSubjects) {
        this.usn = usn;
        this.name = name;
        this.credits = new int[numSubjects];
        this.marks = new int[numSubjects];
    }

    void acceptDetails() {
        System.out.println("Enter USN:");
        usn = s.nextLine();

        System.out.println("Enter Name:");
        name = s.nextLine();

        for (int i = 0; i < credits.length; i++) {
            System.out.println("Enter credits for subject " + (i + 1) + ":");
            credits[i] = s.nextInt();
        }

        for (int i = 0; i < marks.length; i++) {
            System.out.println("Enter marks for subject " + (i + 1) + ":");
            marks[i] = s.nextInt();
        }
    }

    void displayDetails() {
        System.out.println("Details of student:");
        System.out.println("USN = " + usn);
        System.out.println("Name = " + name);

        System.out.println("Credits of subjects:");
        for (int i = 0; i < credits.length; i++) {
            System.out.println("Subject " + (i + 1) + ": " + credits[i]);
        }

        System.out.println("Marks of subjects:");
        for (int i = 0; i < marks.length; i++) {
            System.out.println("Subject " + (i + 1) + ": " + marks[i]);
        }
    }

    double calSGPA() {
```

QUE 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.

```
(base) madhupandey@Madhus-MacBook-Air javalab % javac SGPA.java
(base) madhupandey@Madhus-MacBook-Air javalab % java SGPA
Enter the number of subjects:
3
Enter USN:
1BM22CS140
Enter Name:
Madhu
Enter credits for subject 1:
4
Enter credits for subject 2:
3
Enter credits for subject 3:
1
Enter marks for subject 1:
90
Enter marks for subject 2:
98
Enter marks for subject 3:
97
Details of student:
USN = 1BM22CS140
Name = Madhu
Credits of subjects:
Subject 1: 4
Subject 2: 3
Subject 3: 1
Marks of subjects:
Subject 1: 90
Subject 2: 98
Subject 3: 97
SGPA: 10.0
(base) madhupandey@Madhus-MacBook-Air javalab %
```

```

        for (int i = 0; i < marks.length; i++) {
            System.out.println("Subject " + (i + 1) + ": " + marks[i]);
        }

    double calSGPA() {
        double totalCredit = 0.0, totalGradePoints = 0.0;
        for (int i = 0; i < credits.length; i++) {
            totalCredit += credits[i];
            totalGradePoints += calGradePoints(marks[i]) * credits[i];
        }
        return totalGradePoints / totalCredit;
    }

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

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

            System.out.println("Enter the number of subjects:");
            int numSubjects = scanner.nextInt();

            Students s = new Students("1BM18CS001", "Riya", numSubjects);

            s.acceptDetails();
            s.displayDetails();

            double sgpa = s.calSGPA();
            System.out.println("SGPA: " + sgpa);
        }
    }
}

```

```

method PrintStream.println(char[]) is not applicable
    (actual and formal argument lists differ in length)
method PrintStream.println(String) is not applicable
    (actual and formal argument lists differ in length)
method PrintStream.println(Object) is not applicable
    (actual and formal argument lists differ in length)
[1 error]
(base) madhupandey@Madhus-MacBook-Air javalab % javac SGPA.java
(base) madhupandey@Madhus-MacBook-Air javalab % java SGPA
Enter the number of subjects:
3
Enter USN:
1BM22CS140
Enter Name:
Madhu
Enter credits for subject 1:
4
Enter credits for subject 2:
3
Enter credits for subject 3:
1
Enter marks for subject 1:
90
Enter marks for subject 2:
98
Enter marks for subject 3:
97
Details of student:
USN = 1BM22CS140
Name = Madhu
Credits of subjects:
Subject 1: 4
Subject 2: 3
Subject 3: 1
Marks of subjects:
Subject 1: 90
Subject 2: 98
Subject 3: 97
SGPA: 10.0
(base) madhupandey@Madhus-MacBook-Air javalab %

```

LAB PROGRAM 4

```
BookTest.java Abstract.java + (base) madhupandey@Madhus-MacBook-Air javalab % c  
zsh: command not found: c  
(base) madhupandey@Madhus-MacBook-Air javalab % javac Abstract.java  
(base) madhupandey@Madhus-MacBook-Air javalab % java Abstract  
Rectangle area : = 50  
Triangle area : 12.0  
Circle area : 50.26548245743669  
(base) madhupandey@Madhus-MacBook-Air javalab %  
  
abstract class Shape{  
    int dim1, dim2;  
Shape(int dim1, int dim2){  
    this.dim1=dim1;  
    this.dim2=dim2;  
}  
public abstract void printarea();  
}  
  
class Rectangle extends Shape{  
Rectangle(int length, int width){  
super(length,width);  
}  
public void printarea(){  
    int area=dim1*dim2;  
    System.out.println("Rectangle area : = "+ area);  
}  
}  
  
class Triangle extends Shape{  
Triangle(int base, int height){  
super(base, height);  
}  
public void printarea(){  
    double area=0.5*dim1*dim2;  
    System.out.println("Triangle area : "+ area);  
}  
}  
  
class Circle extends Shape{  
Circle(int radius){  
super(radius,radius);  
}  
public void printarea(){  
    double area=Math.PI*dim1*dim2;  
    System.out.println("Circle area : "+ area);  
}  
}  
  
public class Abstract{  
public static void main(String args[]){  
    Rectangle r= new Rectangle(5,10);  
    Triangle t= new Triangle(3,8);  
    Circle c = new Circle(4);  
  
    r.printarea();  
    t.printarea();  
    c.printarea();  
}
```

QUE 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.

LAB 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.Scanner;

class Account {

    String customerName;
    int accountNumber;
    String accountType;
    double balance;

    Account(String name, int accNo, String accType, double initialBalance) {
        customerName = name;
        accountNumber = accNo;
        accountType = accType;
        balance = initialBalance;
    }

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

```
void displayBalance() {  
    System.out.println("Account Balance: $" + balance);  
}  
}  
  
class SavAcct extends Account {  
    double interestRate;  
  
    SavAcct(String name, int accNo, String accType, double initialBalance, double interest) {  
        super(name, accNo, accType, initialBalance);  
        interestRate = interest;  
    }  
  
    void computeInterest() {  
        double interestAmount = balance * (interestRate / 100);  
        deposit(interestAmount);  
    }  
  
    void withdraw(double amount) {  
        if (balance >= amount) {  
            balance -= amount;  
            System.out.println("Withdrawal of $" + amount + " successful.");  
        } else {  
            System.out.println("Insufficient funds.");  
        }  
    }  
}  
  
class CurAcct extends Account {
```

```
double minimumBalance;
double serviceCharge;

CurAcct(String name, int accNo, String accType, double initialBalance, double
minBalance, double charge) {
    super(name, accNo, accType, initialBalance);
    minimumBalance = minBalance;
    serviceCharge = charge;
}

void checkMinimumBalance() {
    if (balance < minimumBalance) {
        balance -= serviceCharge;
        System.out.println("Service charge of $" + serviceCharge + " applied due to balance
below minimum.");
    }
}

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

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

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

// Create a savings account
System.out.println("Enter name for savings account:");
String savName = scanner.nextLine();
System.out.println("Enter initial balance for savings account:");
double savBalance = scanner.nextDouble();
SavAcct savingsAccount = new SavAcct(savName, 1001, "Savings", savBalance, 5.0);

// Create a current account
System.out.println("Enter name for current account:");
scanner.nextLine(); // Consume newline
String curName = scanner.nextLine();
System.out.println("Enter initial balance for current account:");
double curBalance = scanner.nextDouble();
CurAcct currentAccount = new CurAcct(curName, 2001, "Current", curBalance, 1000.0,
20.0);

// Perform operations on savings account
System.out.println("\nOperations for Savings Account:");
savingsAccount.displayBalance();
System.out.println("Enter amount to deposit in savings account:");
double savDeposit = scanner.nextDouble();
savingsAccount.deposit(savDeposit);
savingsAccount.displayBalance();
savingsAccount.computeInterest();
savingsAccount.displayBalance();
System.out.println("Enter amount to withdraw from savings account:");
double savWithdraw = scanner.nextDouble();
savingsAccount.withdraw(savWithdraw);
savingsAccount.displayBalance();
```

```
// Perform operations on current account  
System.out.println("\nOperations for Current Account:");  
currentAccount.displayBalance();  
System.out.println("Enter amount to deposit in current account:");  
double curDeposit = scanner.nextDouble();  
currentAccount.deposit(curDeposit);  
currentAccount.displayBalance();  
System.out.println("Enter amount to withdraw from current account:");  
double curWithdraw = scanner.nextDouble();  
currentAccount.withdraw(curWithdraw);  
currentAccount.displayBalance();  
  
scanner.close();  
}  
}
```

OUTPUT:

```
C:\Users\Admin\Desktop\CS_140>javac Bank.java  
C:\Users\Admin\Desktop\CS_140>java Bank  
Enter name for savings account:  
Madhu  
Enter initial balance for savings account:  
5000  
Enter name for current account:  
Sarika  
Enter initial balance for current account:  
6000  
  
Operations for Savings Account:  
Account Balance: $5000.0  
Enter amount to deposit in savings account:  
2000  
Deposit of $2000.0 successful.  
Account Balance: $7000.0  
Deposit of $350.0 successful.  
Account Balance: $7350.0  
Enter amount to withdraw from savings account:  
4000  
Withdrawal of $4000.0 successful.  
Account Balance: $3350.0  
  
Operations for Current Account:  
Account Balance: $6000.0  
Enter amount to deposit in current account:  
3000  
Deposit of $3000.0 successful.  
Account Balance: $9000.0  
Enter amount to withdraw from current account:  
8000  
Withdrawal of $8000.0 successful.  
Account Balance: $1000.0  
  
C:\Users\Admin\Desktop\CS_140>
```

QUE 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

LAB PROGRAM 6

```
package CIE;

import java.util.Scanner;

public class Student {
    public String usn;
    public String name;
    public int sem;
    Scanner S1 = new Scanner(System.in);

    public void accept() {
        System.out.println("Enter the USN:");
        usn = S1.next();
        System.out.println("Enter the Name:");
        name = S1.next();
        System.out.println("Enter the Sem:");
        sem = S1.nextInt();
    }

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

```
(base) madhupandey@Madhus-MacBook-Air javalab % javac CIE/Internals.java
(base) madhupandey@Madhus-MacBook-Air javalab % javac SEE/Externals.java
(base) madhupandey@Madhus-MacBook-Air javalab % javac Final.java
(base) madhupandey@Madhus-MacBook-Air javalab % java Final
Enter the value of n
2
Enter the USN:
1BM22CS140
Enter the Name:
Madhu
Enter the Sem:
3
Enter Internal Marks for 5 courses out of 50:
Course 1: 49
Course 2: 50
Course 3: 48
Course 4: 47
Course 5: 46
Enter SEE Marks for 5 courses out of 50:
Course 1: 48
Course 2: 49
Course 3: 48
Course 4: 50
Course 5: 49
Name: Madhu
USN: 1BM22CS140
Sem: 3
Internal Marks: 49 50 48 47 46
SEE Marks: 48 49 48 50 49
Final Marks in Subject 1: 97
Final Marks in Subject 2: 99
Final Marks in Subject 3: 96
Final Marks in Subject 4: 97
Final Marks in Subject 5: 95
Enter the USN:
1BM22CS124
Enter the Name:
Riya
Enter the Sem:
3
Enter Internal Marks for 5 courses out of 50:
Course 1: 34
Course 2: 23
Course 3: 12
Course 4: 32
Course 5: 45
Enter SEE Marks for 5 courses out of 50:
```

```

package CIE;

import java.util.Scanner;

public class Internals extends Student {
    public int[] internalMarks = new int[5];
    Scanner S1 = new Scanner(System.in);

    public void accept() {
        super.accept(); // Call the accept method from the parent class
        to input basic details
        System.out.println("Enter Internal Marks for 5 courses out of
        50:");
        for (int i = 0; i < 5; i++) {
            System.out.print("Course " + (i + 1) + ": ");
            internalMarks[i] = S1.nextInt();
        }
    }

    public void display() {
        super.display(); // Call the display method from the parent class
        to show basic details
        System.out.print("Internal Marks: ");
        for (int i = 0; i < 5; i++) {
            System.out.print(internalMarks[i] + " ");
        }
        System.out.println();
    }
}

```

```

3
Enter Internal Marks for 5 courses out of 50:
Course 1: 49
Course 2: 50
Course 3: 48
Course 4: 47
Course 5: 46
Enter SEE Marks for 5 courses out of 50:
Course 1: 48
Course 2: 49
Course 3: 48
Course 4: 50
Course 5: 49
Name: Madhu
USN: 1BM22CS140
Sem: 3
Internal Marks: 49 50 48 47 46
SEE Marks: 48 49 48 50 49
Final Marks in Subject 1: 97
Final Marks in Subject 2: 99
Final Marks in Subject 3: 96
Final Marks in Subject 4: 97
Final Marks in Subject 5: 95
Enter the USN:
1BM22CS124
Enter the Name:
Riya
Enter the Sem:
3
Enter Internal Marks for 5 courses out of 50:
Course 1: 34
Course 2: 23
Course 3: 12
Course 4: 32
Course 5: 45
Enter SEE Marks for 5 courses out of 50:
Course 1: 43
Course 2: 27
Course 3: 35
Course 4: 41
Course 5: 32
Name: Riya
USN: 1BM22CS124
Sem: 3
Internal Marks: 34 23 12 32 45
SEE Marks: 43 27 35 41 32
Final Marks in Subject 1: 77

```

```
package SEE;

import CIE.Student;

import java.util.Scanner;

public class Externals extends Student {
    public int[] seeMarks = new int[5];
    Scanner S1 = new Scanner(System.in);

    public void accept() {

        System.out.println("Enter SEE Marks for 5 courses out of 50:");
        for (int i = 0; i < 5; i++) {
            System.out.print("Course " + (i + 1) + ": ");
            seeMarks[i] = S1.nextInt();
        }
    }

    public void display() {

        System.out.print("SEE Marks: ");
        for (int i = 0; i < 5; i++) {
            System.out.print(seeMarks[i] + " ");
        }
        System.out.println();
    }
}
```

```
Course 4: 47
Course 5: 46
Enter SEE Marks for 5 courses out of 50:
Course 1: 48
Course 2: 49
Course 3: 48
Course 4: 50
Course 5: 49
Name: Madhu
USN: 1BM22CS140
Sem: 3
Internal Marks: 49 50 48 47 46
SEE Marks: 48 49 48 50 49
Final Marks in Subject 1: 97
Final Marks in Subject 2: 99
Final Marks in Subject 3: 96
Final Marks in Subject 4: 97
Final Marks in Subject 5: 95
Enter the USN:
1BM22CS124
Enter the Name:
Riya
Enter the Sem:
3
Enter Internal Marks for 5 courses out of 50:
Course 1: 34
Course 2: 23
Course 3: 12
Course 4: 32
Course 5: 45
Enter SEE Marks for 5 courses out of 50:
Course 1: 43
Course 2: 27
Course 3: 35
Course 4: 41
Course 5: 32
Name: Riya
USN: 1BM22CS124
Sem: 3
Internal Marks: 34 23 12 32 45
SEE Marks: 43 27 35 41 32
Final Marks in Subject 1: 77
Final Marks in Subject 2: 50
Final Marks in Subject 3: 47
Final Marks in Subject 4: 73
Final Marks in Subject 5: 77
(base) madhupandey@Madhus-MacBook-Air javalab % #
```

```

import CIE.Student;
import CIE.Internals;
import SEE.Externals;

import java.util.Scanner;

public class Final {
    public static void main(String xx[]) {
        Scanner S1 = new Scanner(System.in);
        int n;
        System.out.println("Enter the value of n");
        n = S1.nextInt();

        CIE.Student S[] = new CIE.Student[n];
        CIE.Internals I[] = new CIE.Internals[n];
        SEE.Externals E[] = new SEE.Externals[n];

        for (int i = 0; i < n; i++) {
            S[i] = new CIE.Student();
            I[i] = new CIE.Internals();
            E[i] = new SEE.Externals();

            I[i].accept();
            E[i].accept();

            I[i].display();
            E[i].display();

            for (int j = 0; j < 5; j++) {
                int finalmarks = I[i].internalMarks[j] + E[i].seeMarks[j];
                System.out.println("Final Marks in Subject " + (j + 1) + ":" + finalmarks);
            }
        }
    }
}

```

```

Course 4: 47
Course 5: 46
Enter SEE Marks for 5 courses out of 50:
Course 1: 48
Course 2: 49
Course 3: 48
Course 4: 50
Course 5: 49
Name: Madhu
USN: 1BM22CS140
Sem: 3
Internal Marks: 49 50 48 47 46
SEE Marks: 48 49 48 50 49
Final Marks in Subject 1: 97
Final Marks in Subject 2: 99
Final Marks in Subject 3: 96
Final Marks in Subject 4: 97
Final Marks in Subject 5: 95
Enter the USN:
1BM22CS124
Enter the Name:
Riya
Enter the Sem:
3
Enter Internal Marks for 5 courses out of 50:
Course 1: 34
Course 2: 23
Course 3: 12
Course 4: 32
Course 5: 45
Enter SEE Marks for 5 courses out of 50:
Course 1: 43
Course 2: 27
Course 3: 35
Course 4: 41
Course 5: 32
Name: Riya
USN: 1BM22CS124
Sem: 3
Internal Marks: 34 23 12 32 45
SEE Marks: 43 27 35 41 32
Final Marks in Subject 1: 77
Final Marks in Subject 2: 50
Final Marks in Subject 3: 47
Final Marks in Subject 4: 73
Final Marks in Subject 5: 77
(base) madhupandey@Madhus-MacBook-Air javalab % #

```

LAB 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.

```
import java.util.Scanner;

// Custom exception class WrongAge
class WrongAge extends Exception {
    WrongAge() {
        super("Invalid age. Age cannot be less than 0.");
    }

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

// Base class Father
class Father {
    private int age;

    // Constructor in Father class with age validation
    public Father(int age) throws WrongAge {
        if (age < 0) {
            throw new WrongAge();
        }
        this.age = age;
    }

    public int getAge() {
        return age;
    }
}

// Derived class Son extending Father
class Son extends Father {
    private int sonAge;

    // Constructor in Son class with additional age validation
    public Son(int fatherAge, int sonAge) throws WrongAge {
        super(fatherAge); // Call the constructor of the base class

        if (sonAge < 0) {
            throw new WrongAge("Son's age cannot be less than 0.");
        } else if (sonAge >= fatherAge) {
            throw new WrongAge("Son's age should be less than Father's
age.");
        }
    }

    this.sonAge = sonAge;
```

```
}

public int getSonAge() {
    return sonAge;
}
}

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

        try {
            // Taking user input for Father's age
            System.out.print("Enter Father's age: ");
            int fatherAge = scanner.nextInt();
            Father father = new Father(fatherAge);
            System.out.println("Father's age: " + father.getAge());

            // Taking user input for Son's age
            System.out.print("Enter Son's age: ");
            int sonAge = scanner.nextInt();
            Son son = new Son(father.getAge(), sonAge);
            System.out.println("Son's age: " + son.getSonAge());

        } catch (WrongAge e) {
            System.out.println("Exception: " + e.getMessage());
        }
    }
}
```

Enter Father's age: 32

Father's age: 32

Enter Son's age: 5

Son's age: 5

Process finished with exit code 0

LAB 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) {  
            System.out.println(message);  
            try {  
                Thread.sleep(interval * 1000); // Convert seconds to milliseconds  
            } catch (InterruptedException e) {  
                System.out.println("Exception :" + e);  
            }  
        }  
    }  
}  
  
public class DisplayMessages {  
    public static void main(String[] args) {  
        DisplayThread thread1 = new DisplayThread("BMS College of Engineering", 10);  
        DisplayThread thread2 = new DisplayThread("CSE", 2);  
  
        thread1.start();  
        thread2.start();  
    }  
}
```

```
(base) madhupandey@Madhus-MacBook-Air ~ % cd desktop
(base) madhupandey@Madhus-MacBook-Air desktop % cd javamulti
(base) madhupandey@Madhus-MacBook-Air javamulti % javac displaymessages.java
(base) madhupandey@Madhus-MacBook-Air javamulti % java DisplayMessages
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
```

LAB 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 ArithmeticException. 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 DivisionUI extends JFrame {

    private JTextField num1Field, num2Field, resultField;

    public DivisionUI() {
        setTitle("Integer Division Calculator");
        setSize(300, 150);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocationRelativeTo(null);

        createUI();
    }

    private void createUI() {
        JPanel panel = new JPanel();
        panel.setLayout(new GridLayout(4, 2));

        JLabel num1Label = new JLabel("Num1:");
        num1Field = new JTextField();
        JLabel num2Label = new JLabel("Num2:");
        num2Field = new JTextField();
        JLabel resultLabel = new JLabel("Result:");
        resultField = new JTextField();
        resultField.setEditable(false);

        JButton divideButton = new JButton("Divide");
        divideButton.addActionListener(new ActionListener() {

            public void actionPerformed(ActionEvent e) {
                performDivision();
            }
        });
    }
}
```

```

        panel.add(num1Label);
        panel.add(num1Field);
        panel.add(num2Label);
        panel.add(num2Field);
        panel.add(resultLabel);
        panel.add(resultField);
        panel.add(new JLabel()); // Empty label for spacing
        panel.add(divideButton);

        add(panel);
    }

private void performDivision() {
    try {
        int num1 = Integer.parseInt(num1Field.getText());
        int num2 = Integer.parseInt(num2Field.getText());

        if (num2 == 0) {
            throw new ArithmeticException("Division by zero is not allowed");
        }

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

    } catch (NumberFormatException e) {
        JOptionPane.showMessageDialog(this, "Please enter valid integers for Num1 and
Num2.",
        "Number Format Error", JOptionPane.ERROR_MESSAGE);

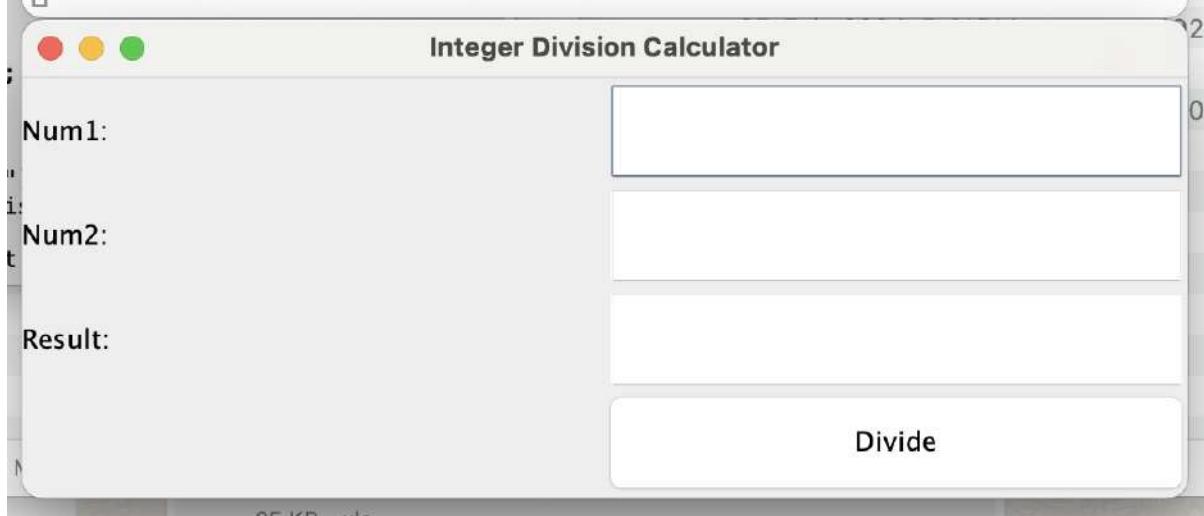
    } catch (ArithmeticException e) {
        JOptionPane.showMessageDialog(this, "Cannot divide by zero.", "Arithmetic Error",
        JOptionPane.ERROR_MESSAGE);
    }
}

public static void main(String[] args) {
    SwingUtilities.invokeLater(new Runnable {

        public void run() {
            new DivisionUI().setVisible(true);
        }
    });
}
}

```

```
[(base) madhupandey@Madhus-MacBook-Air desktop % javac DivisionUI.java  
[(base) madhupandey@Madhus-MacBook-Air desktop % java DivisionUI.java
```



LAB PROGRAM-1

DATE: 18/12/23 PAGE: 1

Quadratic equation

```
import java.util.Scanner;
```

```
class Quad
```

```
{
```

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

```
{ double a, b, c, d, x1, x2;
```

```
Scanner s1 = new Scanner(System.in);
```

```
a = s1.nextDouble();
```

```
b = s1.nextDouble();
```

```
c = s1.nextDouble();
```

```
if (a == 0 || b == 0 || c == 0) {
```

```
System.out.println("Invalid Input");
```

```
}
```

```
else {
```

```
d = b * b - 4 * a * c;
```

```
if (d > 0) {
```

```
System.out.println("Roots are real and equal");
```

```
x1 = (-b + Math.sqrt(d)) / (2 * a);
```

```
x2 = (-b - Math.sqrt(d)) / (2 * a);
```

```
System.out.println("x1" + " = " + x1 + " and " + "x2" +  
" = " + x2);
```

```
}
```

```
if (d < 0) {
```

```
System.out.println("Roots are imaginary and distinct");
```

```
x1 = -b / (2 * a);
```

```
x2 = Math.sqrt(-d) / (2 * a);
```

```
System.out.println("x1" + " = " + x1 + " and " + "x2" +  
" = " + x2);
```

```
}
```

```
else {
```

```
System.out.println("Roots are equal");
```

```
x1 = x2 = -b / (2 * a);
```

```
System.out.println("x1" + " and " + "x2" + " = " + x1);
```

```
}
```

```
}
```

Output:-

2 4 4

Roots are Imaginary. ~~and diff~~

$\alpha_1 = -4.0$ and $\alpha_2 = \text{NAN}$

2 4 2

Roots are equal

α_1 and $\alpha_2 = -1.0$

1 5 2

Roots are real and distinct.

Page 2. class First {

```
public static void main(String args[]) {
    System.out.println("Hello World");
}
```

3

Output:-

Hello World,

- Q. Write a Java program to create a class Employee with members empid, empname, empnohours, empbasic, emphra(%), empda(%), empft(%), empgross.
- Include methods to do the following:
- Accept all the values from the user. Note HRA, DA and IT are given in %.
 - Calculate the gross salary based on formula.

$$\text{empgross} = \text{empbasic} + \text{empbasic} * \text{emphra} + \text{empbasic} * \text{empda} - \text{empbasic} * \text{empft}$$
 - Consider the overtime amount to be Rs. 100 per hour. If empnohours > 200, for every hour the employee has to be given additional payment. Calculate the additional payment and update the gross. If empnohours < 200, reduce Rs. 100 per hour and update the gross.

Import java.util.Scanner;

class Employee {

```
public static void main(String args[])
{
    int empid, empnohours, empbasic;
    String empname;
    float emphra, empda, empft, empgross;
    Scanner s1 = new Scanner(System.in);
    System.out.println("enter employee name");
    empname = s1.next();
    System.out.println("enter employee id");
    empid = s1.nextInt();
    System.out.println("enter no of hrs");
    empnohours = s1.nextInt();
    System.out.println("enter empbasic");
    s1.nextInt();
    empbasic = s1.nextInt();
    System.out.println("enter emphra");
    empft = s1.nextFloat();
    System.out.println("enter empda");
    empda = s1.nextFloat();
    System.out.println("enter emphra");
    empgross = s1.nextFloat();
}
```

```
System.out.println("Enter empftft");
```

```
empft = sc.nextInt();
```

```
empgross = empbasic + empbasic * empft + empbasic * empda  
- empbasic * empft;
```

```
System.out.println(" empgross : " + empgross);
```

```
if (empft > 200) {
```

```
double overtimePayment = (empft - 200) * 100;
```

```
empgross += overtimePayment;
```

```
System.out.println(" overtime payment :- " + overtimePay  
- ment);
```

```
System.out.println(" Updated empgross : " + empgross);
```

```
else if (empft < 200) {
```

```
double deduction = (empft - 200 - empft) * 100;
```

```
empgross -= deduction;
```

```
System.out.println(" Deduction :- " + deduction);
```

```
System.out.println(" Updated empgross : " + empgross);
```

}

Outputs —

Enter employee name

Madhu

Overtime Payment: 1000

Updated empgross :- 112000

Enter employee id

1

Enter no of hrs

910

Enter empbasic

20000

Enter empfta

40

Enter empda

35

Enter empft

20

empgross:- 1120000

LAB PROGRAM - 2

DATE:

PAGE:

```
import java.util.Scanner;  
class Students {  
    String usn, name;  
    int [] credits;  
    int [] marks;  
    Scanner s = new Scanner(System.in);
```

```
Students(String usn, String name, int [] credits, int [] marks){  
    this.usn = usn;  
    this.name = name;  
    this.credits = credits;  
    this.marks = marks;  
}
```

```
void acceptDetails(){  
    credits = new int[marks.length];  
    System.out.println("Enter usn");  
    usn = s.nextLine();  
    System.out.println("Enter name");  
    name = s.nextLine();
```

```
System.out.println("Enter credits");  
for(int i=0; i<marks.length; i++){
```

```
    System.out.println("Enter credits of sub"+(i+1)+":  
    + credits[i]);
```

}

```
System.out.println("Enter marks");
```

```
for(int i=0; i<credits.length; i++){
```

```
    System.out.println("Enter marks of subject"+  
    (i+1)+":");
```

```
    marks[i] = s.nextInt();
```

}

}

```

void displayDetails() {
    System.out.println(" Details of student");
    System.out.println(" USN = " + usn);
    System.out.println(" Name= " + name);
}

{
    System.out.println(" Marks of subjects");
    for(int i=0; i< marks.length; i++) {
        System.out.println(" Subject " + (i+1) + ":" +
                           + credits[i]);
    }
}

System.out.println(" Marks of subjects");
for(int i=0; i< marks.length; i++) {
    System.out.println(" Subject " + (i+1) + ":" +
}
}

double calSGPA() {
    double totalCredit=0.0, totalGradePoints=0.0;
    for(int i=0; i< credits.length; i++) {
        totalCredit += calGradePoints(marks[i]) *
                      credits[i];
    }
    return totalGradePoints / totalCredit;
}

int calGradePoints(int marks) {
    if (marks >= 90) {
        return 10;
    } else if (marks >= 80) {
        return 9;
    } else if (marks >= 70) {
        return 8;
    } else if (marks >= 60) {
        return 7;
    }
}

```

```

else if (marks >= 50) {
    between 6;
} else if (marks >= 40) {
    between 5;
} else {
    between 0;
}
}

```

```

class & SGPA {

```

```

public static void main(String args[]) {
    int [] credits = {4, 3, 3, 2};
    int [] marks = {85, 75, 90, 60};
    Students s = new Students("IBM22CS120", "Riya", credits,
        marks);
    s.acceptDetails();
    s.displayDetails();
    double sgpa = s. calculateSGPA();
    System.out.println("SGPA :" + SGPA);
}

```

Output:-

Enter usn:

IBM22CS120

Enter name:

Riya

Enter credits:

Enter credits of sub1

4

Enter credits of sub2

3

Enter credits of sub 3:

3

Enter credits of sub 4:

2

Display details :-

USN = 1bm22cs140

Name = matbu

Credit of subjects:

Subject 1: 4

Subject 2: 3

Subject 3: 3

Subject 4: 2

Marks of subjects:

Subject 1: 90

Subject 2: 98

Subject 3: 97

Subject 4: 96

SGPA: 10.00

V
11124

LAB PROGRAM - 3

DATE:

PAGE:

```
import java.util.Scanner;  
class Book {  
    String name;  
    String 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;  
}
```

```
void setName(String name) {  
    this.name = name;  
}
```

```
String getName() {  
    return name;  
}
```

```
void setAuthor() {  
    this.author =  
    return author;  
}
```

```
String getAuthor() {  
    return author;  
}
```

```
void setPrice(int price) {  
    this.price = price;  
}
```

```
int getPrice() {  
    return price;  
}
```

int num-pages

DATE:

PAGE:

void setnum-pages() {

this.num-pages = num-pages

}

int getnum-pages() {

return num-pages;

}

String toString() {

return "Book Details: \n Name: " + name + "\n Author:

+ author + "\n Price: " + price + "\n Number of pages:

+ num-pages";

}

class BookTest {

public static void main(String args[]) {

System.out.println("Enter number of books to create:");

Scanner sc = new Scanner(System.in);

int n = sc.nextInt();

Books[] books = new Book[n];

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

System.out.println("Enter details of book" + (i + 1) + ":");

System.out.println("Name: ");

String name = sc.next();

System.out.println("Price: ");

int price = sc.nextInt();

System.out.println("Num of pages");

int num-pages = sc.nextInt();

books[i] = new Book(name, author, price, num-pages);

System.out.println("Above values are " + books[i].toString());

}

}

11/11/21

LAB PROGRAM-4

DATE:

PAGE:

abstract class Shape {

int dim1, dim2;

Shape(int dim1, int dim2) {

this.dim1 = dim1;

this.dim2 = dim2;

}

public abstract void paintarea();

}

class Rectangle extends Shape {

Rectangle(int length, int width) {

super(length, width);

}

public void paintarea() {

int area = dim1 * dim2;

System.out.println("Rectangle area = " + area);

}

class Triangle extends Shape {

Triangle(int base, int height) {

super(base * height);

public void paintarea() {

double area = 0.5 * dim1 * dim2;

System.out.println("Triangle : " + area);

}

class Circle extends Shape {

Circle(int radius) {

super(radius, radius);

public void paintarea() {

```
double area = Math.PI * dim1 * dim2;  
System.out.println("Circle " + area);
```

3
public class Abst{

```
public static void main(String args[]){  
    Rectangle r = new Rectangle(5,10);  
    Triangle t = new Triangle(3,18);  
    Circle c = new Circle(4);
```

r.printarea();
t.printarea();
c.printarea();

3

Output:-

Rectangle area = 50

Triangle area = 12.0

Circle area = 50.265482

✓ 8/17/24

LAB Program 6

DATE:

PAGE:

Create a package CIE which has two classes - Student and Internals. The class Student 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 SEE marks scored in five courses of the current semester of the student. Import the two packages in a file that declares the final marks of n students in all five courses.

```
package cie;
import java.util.Scanner;
public class Student {
    protected String usn;
    protected String name;
    protected int sem;
    public void acceptDetails() {
        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 displayDetails() {
        System.out.println("USN : " + usn);
        System.out.println("Name : " + name);
        System.out.println("Semester : " + sem);
    }
}
```

package cie;

import java.util.Scanner; ^{cie}
public class Internal extends Student

protected int[] internalMarks;

// Method to accept details for internal
public void acceptDetails() {

Scanner s = new Scanner(System.in);

internalMarks = new int[5];

System.out.println("Enter the internal marks for
for (int i=0; i<5; i++) {

System.out.println("Course" + (i+1) + ":");

internalMarks[i] = s.nextInt();

}

public void displayDetails() {

System.out.println("Internal Marks");

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

System.out.println("Course" + (i+1) + "
+ internalMarks[i]);

}

}

package SEE;

import cie.Student;

import java.util.Scanner;

public class External extends Student {

protected int[] seeMarks;

public void acceptDetails() {

super.acceptDetails(); // calls the method from sub-class Student.

Scanner s = new Scanner(System.in);

seeMarks = new int[5]; ^{Enter SEE marks for five courses}

System.out.println("Course" + (i+1) + ":");

seeMarks[i] = s.nextInt();

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

```
System.out.println("Course " + (i+1) + ": ");
```

```
see Marks[i] = s.nextInt();
```

```
}
```

{

}

```
super.displayDetails();
```

```
System.out.println(" SEE Marks");
```

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

```
System.out.println("Course " + (i+1) + ": " + seeMarks[i]);
```

```
}
```

{

}

```
import CIE.Student;
```

```
import CIE.Internal;
```

```
import SEE.External;
```

```
public class calculateFinalMarks {
```

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

```
        Scanner s1 = new Scanner(System.in);
```

```
        int n;
```

```
        System.out.println("Enter the value of n");
```

```
        n = s1.nextInt();
```

```
        CIE.Student S[] = new CIE.Student[n];
```

```
        CIE.Internal I[] = new CIE.Internal[n];
```

```
        SEE.External ET[] = new SEE.External[n];
```

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

```
            S[i] = new CIE.Student();
```

```
            I[i] = new CIE.Internal();
```

```
            E[i] = new SEE.External();
```

```
            S[i].acceptDetails(); I[i].acceptDetails();
```

```
            E[i].acceptDetails(); S[i].displayDetails();
```

```
            I[i].displayDetails(); ET[i].displayDetails();
```

```
            for (int j=0; j<5; j++) {
```

```
                int finalMarks = I[i].JMarks[j] + ET[i].EMarks[j];
```

```
                System.out.println(" Marks in " + (j+1) + " Subject " + finalMarks);
```

```
}
```

{

}

LAB Program #7

DATE: 21/1/24 PAGE:

Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class after called "Father" and derived class called "Son" which extends the base class. In a father class, implement a constructor which takes the age and throws the exception WrongAge(). when input age < 0. In a son class, implement a constructor that cases both father and son's age and throws an exception if son's age is \geq father's age.

import java.util.Scanner;

class WrongAge extends Exception
WrongAge() {

} Super("Invalid Age. Age can not be less than zero")

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

}

class Father {

private int age;

public Father(int age) throws WrongAge {
if (age < 0) {

throw new WrongAge();
}

this.age = age;
}

public int getAge() {
return age;
}

class Son extends Father {

private int sonAge;

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

super(fatherAge); // calls the constructor
of the class Father.

if (sonAge < 0) {

throw new WrongAge ("Son's age can't
be less than 0.");

} else if (sonAge >= fatherAge) {

throw new WrongAge ("Son's age should
be less than father's age");

this.sonAge = sonAge;

public int getSonAge() {

return sonAge;

public class Main {

public static void main (String args[]) {

Scanner os = new Scanner (System.in);

key {

System.out.println ("Enter father's age");

int fatherAge = scanner.nextInt();

Father father = new Father (fatherAge);

System.out.println ("Father's age: " +

father.getAge());

System.out.println("Enter son's age");
 int sonAge = s.nextInt();

Son son = new Son(father.getAge(), sonAge);
 System.out.println("Son's age: " + son.getSonAge());

} catch (WrongAge e) {

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

// getMessage is a method
 that is defined by the class.

Output:-

Enter Father's age: 32

Father's Age: 32

Enter Son's age: 5

Son's age: 5

✓ 21/12/24

import java.util.Scanner;

class WrongSal extends Exception {
 WrongSal() {
 super("invalid Salary");
 }

WrongSal(String message) {
 super(message); }] constructor

{

class Manager {

private int sal;

public Manager(int sal) throws new WrongSal {
 if(sal < 0) {

throw new WrongSal(); }]

this.sal = sal; }]

public int getSal() {

return sal; }]

{

class Worker extends Manager {

private int workerSal; sal

public Worker(int managerSal, int workerSal) throws
new WrongSal {

super(managerSal); // on passing manager sal
of will call constructor.

manager of Manager class to
check for sal.

if (WorkerSal < 0) {

throw new WrongSal("Salary can not be less than zero");

else if (managerSal < workerSal) {

throw new WrongSal("Manager's salary cannot be less than worker's salary.");

}

this.workerSal = workerSal;

}

public int getWorkerSal() {
return workerSal;

}

public class MainClass {

public static void main (String args[]) {

Scanner s = new Scanner (System.in);

try {

s. o. p ("Enter manager's Salary : ");

int managerSal = s.nextInt();

Manager manager = new Manager (managerSal);

s. o. p ("Manager's salary: " + manager.getSal());

s. o. p ("Enter Worker's Salary : ");

int workerSal = s.nextInt();

Worker worker = new Worker

manager.getSal(), workerSal);

} catch (WrongSal e) {

s. o. p ("exception : " + e.getMessage());

Output —

Enter manager's salary: 30000

Enter worker's salary: 34000

Exception: Manager's salary can't be less than
worker's salary.

class MyGenericClass<T, U> {

private T firstValue;

private U secondValue;

MyGenericClass(T firstValue, U secondValue) {

this.firstValue = firstValue;

this.secondValue = secondValue;

}

public T getFirstValue() {

return firstValue;

public U getSecondValue() {

return secondValue;

~~return~~ public void displayType() {

S.O.P("Type of first value: " + firstValue.getClass().
getName());

S.O.P("Type of second value: " + secondValue.getClass().
getName());

public class Generics {

~~public~~ void Steing args[] {

MyGenericClass genPair<Integer, Double> = new MyGenericClass(1, 1.2);

S.O.P("First value(Integer): " + genPair.getFirstValue());

S.O.P("Second value(Double): " + genPair.getSecondValue());

LAB PROGRAM

DATE:

PAGE:

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) {

System.out.println(message);

try {

Thread.sleep(interval * 1000);

} catch (InterruptedException e) {

S.O.P ("Exception: " + e);

public class DisplayMessages {

public static void main (String [] args) {

DisplayThread thread1 = new DisplayThread

("BMS College of Engineering", 10);

DisplayThread thread2 = new DisplayThread ("CSE", 2);

thread1.start();

thread2.start();

Output:-

* BMS College of Engineering

CSE

CSE</

LAB PROGRAM - 5

```
import java.util.Scanner;
```

```
class Account {
```

```
    String customerName;
```

```
    int accNo;
```

```
    String accType;
```

```
    double balance;
```

```
    Account (String name, int accNum, String accountType,  
            double initialBalance) {
```

```
        customerName = name;
```

```
        accNo = accNum;
```

```
        accType = accountType;
```

```
        balance = initialBalance;
```

```
}
```

```
    void deposit (double amount) {
```

```
        balance += amount;
```

```
        System.out.println ("Deposit of " + amount + " successful");
```

```
}
```

```
    void displayBalance () {
```

```
        System.out.println ("Account Balance : " + balance);
```

```
}
```

```
class Savacc extends Account {
```

```
    double interestRate;
```

```
Savacc (String name, int accNum, String accountType,  
        double initialBalance, double interest) {
```

```
    super (name, accNum, accountType, initialBalance);
```

```
    interestRate = interest;
```

```
}
```

```
void compoundInterest() {
```

```
    double interestAmount = balance * (interestRate / 100);  
    deposit(interestAmount);
```

{

```
void withdraw(double amount) {
```

```
    if (balance >= amount) {
```

```
        balance -= amount;
```

```
        System.out.println("Withdrawal of " + amount +
```

```
        " was successful");
```

```
    } else {
```

```
        System.out.println("Insufficient funds");
```

{

{

```
class Current extends Account {
```

```
    double minimumBalance;
```

```
    double serviceCharge;
```

```
Current(String name, int accNum, String accountType,  
        double initialBalance, double minBalance,  
        double charge) {
```

```
    super(name, accNum, accountType, initialBalance);
```

```
    minimumBalance = minBalance;
```

```
    serviceCharge = charge;
```

```
}
```

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

```

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

```

public class Bank {

```

public static void main(String[] args) {
    Scanner s = new Scanner(System.in);
}

```

// Create a savings account

```

SOP("Enter name for savings account:");
String savName = s.nextLine();
SOP("Enter initial balance for savings account:");
double savBalance = s.nextDouble();
SavAcct.savingsAccount = new SavAcct(savName,
                                      "Savings", savBalance, 5.0);

```

// Create a current account

```

SOP("Enter name for current account:");
s.nextLine(); // consume newline
String curName = s.nextLine();
SOP("Enter initial balance for current account:");
double curBalance = scanner.s.nextDouble();
CurAcct.currentAccount = new CurAcct(curName,
                                      "Current", curBalance, 1000.0, 20.0);

```

// Perform operations on savings account,
 S.o.P ("In operations for savings account : ");
 Savings Account . displayBalance ();

S.o.P ("Enter amount to deposit in savings account : ");
 double savDeposit = &s. nextDouble ();
 Savings Account . deposit (savDeposit);
 Savings Account . displayBalance ();
 Savings Account . computeInterest ();
 Savings Account . displayBalance ();

S.o.P ("Enter amount to withdraw from savings
 account : ");

double savWithDraw = &s. nextDouble ();
 Savings Account . withdraw (savWithDraw);
 Savings Account . displayBalance ();

// Perform operations on current account

S.o.P ("In operations for current account : ");
 Current Account . displayBalance ();

S.o.P ("Enter amount to deposit in current account : ");
 double curDeposit = &s. nextDouble ();
 Current Account . deposit (curDeposit);

Current Account . displayBalance ();
 S.o.P ("Enter amount to withdraw from current
 account : ");

double curWithDraw = &s. nextDouble ();
 Current Account . withdraw (curWithDraw);
 Current Account . displayBalance ();

{

} In the below code understand the program with

Output:

: Enter name for savings account : .

Madhuri

: Enter initial balance for savings account : .

5000

: Enter name for current account : .

Fauzia

: Enter initial balance for current account : .

6000

Operations on savings account :

Account Balance : 5000.0

: Enter amount to deposit in savings account : .

2000

Deposit of 2000 successful.

Account Balance : 7000

Deposit of 350 successful.

Account Balance : 7350.0

: Enter amount to withdraw from savings account : .

4000

Withdrawal of 4000 successful.

Account Balance : 3350.0

Operations on current Account :

Account Balance : 6000.0

: Enter amount to deposit in current account : .

3000

Deposit of 3000 unsuccessful

Account Balance : 9000.0

: Enter amount to withdraw from current account : .

3000.0

Withdrawal of 3000.0 successful.

Account Balance 1000.0

LAB - PROGRAM 9

DATE:

PAGE:

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

```
public class EventDemo extends JFrame implements  
ActionListener {
```

```
 JTextField t1, t2, t3;
```

```
JLabel l1, l2;
```

```
JButton b1, b2;
```

```
Public EventDemo {
```

```
setLayout(new FlowLayout());
```

```
l1 = new JLabel("Num 1:");
```

```
add(l1);
```

```
t1 = new JTextField(5);
```

```
add(t1);
```

```
l2 = new JLabel("Num 2:");
```

```
add(l2);
```

```
t2 = new JTextField(5); add(t2);
```

```
t3 = new JTextField(5);
```

```
t3.setEditable(false);
```

```
add(t3);
```

```
b1 = new JButton("Divide");
```

```
add(b1);
```

```
b1.addActionListener(this);
```

```
b2 = new JButton("clear");
```

```
add(b2);
```

```
b2.addActionListener(this);
```

```
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
setSize(300, 150);
```

```
setVisible(true);
```

```
}
```

```
public void actionPerformed(ActionEvent e) {
```

```
    SwingUtilities.invokeLater(new Runnable() {
```

```
        public void run() {
```

```
            if (e.getSource() == button) {
```

```
                int num1 = Integer.parseInt(textField1.getText());
```

```
                int num2 = Integer.parseInt(textField2.getText());
```

```
                int num3 = num1 / num2;
```

```
                textField3.setText(" " + num3);
```

```
            } catch (ArithmaticException e) {
```

```
                JOptionPane.showMessageDialog(this,
```

```
                    "Arithmatic Exception");
```

```
            } catch (NumberFormatException e) {
```

```
                JOptionPane.showMessageDialog(this, "Number  
entered is : Invalid");
```

```
        } else {
```

```
            textField1.setText(" ");
```

```
            textField2.setText(" ");
```

```
            textField3.setText(" ");
```

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

```
        SwingUtilities.invokeLater(new Runnable() {
```

```
            public void run() {
```

```
                new EventDemo();
```

Output:-

num1:	12
-------	----

num2:	6	2
-------	---	---

Divide	Clear
--------	-------

Report

1. ButtonDemo

In this program, java file is first compiled and later the applet code is saved in another text file with html extension. It is then compiled to obtain the output as follows. It opens a window with three buttons yes, no and undecided. As these buttons are clicked messages are displayed in that window.

2. ButtonJiggle

On executing the above program a small window appears with number blocks from 1 to 9 along with reset, start and restart button. When the number block (any one block) is clicked it gets swapped with other block. This happens after clicking the start button. Once the reset button is clicked the blocks get up again from first.

3. ButtonList

On compiling the above programs a pop up window appears with yes, no and undecided buttons. On clicking these buttons a message is being displayed on the same window just below these three buttons.

4. ButtonListD

The only difference between the 4th and 3rd program is that once any of these three buttons are clicked a new window opens and a message is displayed on that window.

5. Division main:

Once this program is compiled Division of Integers window appears with three text fields named num1, num2 and the other with the result. Once two numbers are entered in each text field the result along with two numbers is displayed as floating point numbers.

6. Division main:

This program is similar to the above but the only difference is in the result the modulus value is only displayed along with the two numbers num1 and num2.

7. JTextFieldDemo

On compiling the above program a window named IF-labelDemo appears with two textfields Name and Password along with these name selected text in name and password is displayed in the same window.