



Department of Computer science Engineering
Engineering

B. M. S. COLLEGE OF ENGINEERING

(Accredited by NBA, under Tier 1, 2014-2022)

(Autonomous Institution Affiliated to Visvesvaraya Technological University,
Belagavi) PB 1908, Bull Temple Road, Bengaluru – 560 019

AAT REPORT

Submitted by

AISHWARYA VIJAY

Roll no. :

1BM22CS025

Course Instructor

Prof. Swati Sridharan

DEPARTMENT OF COMPUTER
SCIENCE & ENGINEERING

Lab 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 discriminant b^2-4ac is negative, display a message stating that there are no real solutions.

```
import java. util.Scanner;

public class QuadraticEquationSolver {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.println("Enter the coefficients of the quadratic equation (ax^2 + bx + c = 0):");

        System.out.print("Enter a: ");

        double a = scanner.nextDouble();

        System.out.print("Enter b: ");

        double b = scanner.nextDouble();

        System.out.print("Enter 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("Two real solutions exist:");

            System.out.println("Root 1 = " + root1);

            System.out.println("Root 2 = " + root2);

        } else if (discriminant == 0) {

            double root = -b / (2 * a);

            System.out.println("One real solution exists:");

            System.out.println("Root = " + root);

        } else {

            System.out.println("No real solutions exist.");

        }

        scanner.close();

    }

}
```

}

```
C:\Users\Aishwarya Vijay\Desktop\1BM22CS025>java QuadraticEquationSolver
Enter the coefficients of the quadratic equation (ax^2 + bx + c = 0):
Enter a: 1
Enter b: 0
Enter c: 5
No real solutions exist.

C:\Users\Aishwarya Vijay\Desktop\1BM22CS025>javac QuadraticEquationSolver.java

C:\Users\Aishwarya Vijay\Desktop\1BM22CS025>java QuadraticEquationSolver
Enter the coefficients of the quadratic equation (ax^2 + bx + c = 0):
Enter a: 22.8
Enter b: 7.9
Enter c: 7
No real solutions exist.

C:\Users\Aishwarya Vijay\Desktop\1BM22CS025>javac QuadraticEquationSolver.java

C:\Users\Aishwarya Vijay\Desktop\1BM22CS025>java QuadraticEquationSolver
Enter the coefficients of the quadratic equation (ax^2 + bx + c = 0):
Enter a: 2
Enter b: 4
Enter c: 2
One real solution exists:
Root = -1.0
```

Lab 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 usn;
    String name;
    int[] credits;
    int[] marks;

    // Method to accept details of the student
    void acceptDetails() {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter USN:");
        usn = scanner.nextLine();
        System.out.println("Enter Name:");
        name = scanner.nextLine();
        System.out.println("Enter the number of subjects:");
        int numSubjects = scanner.nextInt();

        // Initialize credits and marks arrays
        credits = new int[numSubjects];
        marks = new int[numSubjects];

        // Input credits and marks for each subject
        for (int i = 0; i < numSubjects; i++) {
            System.out.println("Enter credits for subject " + (i + 1) + ":");
            credits[i] = scanner.nextInt();

            System.out.println("Enter marks for subject " + (i + 1) + ":");
            marks[i] = scanner.nextInt();
        }
    }
}
```

```
// Method to display details of the student
```

```
void displayDetails() {  
    System.out.println("USN: " + usn);  
    System.out.println("Name: " + name);  
    System.out.println("Subject-wise Details:");  
    for (int i = 0; i < credits.length; i++) {  
        System.out.println("Subject " + (i + 1) + ": Credits - " + credits[i] + ", Marks - " + marks[i]);  
    }  
}
```

```
// Method to calculate SGPA
```

```
double calculateSGPA() {  
    int totalCredits = 0;  
    double totalGradePoints = 0.0;  
    for (int i = 0; i < credits.length; i++) {  
        totalCredits += credits[i];  
        totalGradePoints += calculateGradePoint(marks[i]) * credits[i];  
    }  
    return totalGradePoints / totalCredits;  
}
```

```
// Method to calculate grade point based on marks
```

```
double calculateGradePoint(int marks) {  
    if (marks >= 90)  
        return 10.0;  
    else if (marks >= 80)  
        return 9.0;  
    else if (marks >= 70)  
        return 8.0;  
    else if (marks >= 60)  
        return 7.0;  
    else if (marks >= 50)  
        return 6.0;  
    else if (marks >= 40)
```

```
        return 5.0;
    else
        return 0.0;
    }
}
```

```
public class SGPA {
    public static void main(String[] args) {
        Student student = new Student();
        student.acceptDetails();
        System.out.println("Details of the student:");
        student.displayDetails();
        System.out.println("SGPA: " + student.calculateSGPA());
    }
}
```

```
C:\Users\Aishwarya Vijay\Desktop\1BM22CS025>java SGPA
Enter USN:
1BM22CS025
Enter Name:
AISHWARYA VIJAY
Enter the number of subjects:
4
Enter credits for subject 1:
4
Enter marks for subject 1:
84
Enter credits for subject 2:
4
Enter marks for subject 2:
90
Enter credits for subject 3:
3
Enter marks for subject 3:
94
Enter credits for subject 4:
3
Enter marks for subject 4:
90
Details of the student:
USN: 1BM22CS025
Name: AISHWARYA VIJAY
Subject-wise Details:
Subject 1: Credits - 4, Marks - 84
Subject 2: Credits - 4, Marks - 90
Subject 3: Credits - 3, Marks - 94
Subject 4: Credits - 3, Marks - 90
SGPA: 9.714285714285714
```

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;

class Book {
    String name;
    String author;
    double price;
    int num_pages;
    // Constructor to set the values for the members
    Book(String name, String author, double price, int num_pages) {
        this.name = name;
        this.author = author;
        this.price = price;
        this.num_pages = num_pages;
    }
    // Method to set details of the book
    void setDetails(String name, String author, double price, int num_pages) {
        this.name = name;
        this.author = author;
        this.price = price;
        this.num_pages = num_pages;
    }
    // Method to get details of the book
    String getDetails() {
        return "Name: " + name + ", Author: " + author + ", Price: $" + price + ", Number of Pages: " +
num_pages;
    }
}
```



```

// Method to display complete details of the book
String toStringMethod() {
    return "Book Details:\n" + getDetails();
}
}

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

        Book[] books = new Book[n]; // Array to store n book objects
        // Input details for each book
        for (int i = 0; i < n; i++) {
            System.out.println("\nEnter details for Book " + (i + 1) + ":");
            System.out.print("Name: ");
            String name = scanner.nextLine();
            System.out.print("Author: ");
            String author = scanner.nextLine();
            System.out.print("Price: $");
            double price = scanner.nextDouble();
            System.out.print("Number of Pages: ");
            int numPages = scanner.nextInt();
            scanner.nextLine(); // Consume newline

            books[i] = new Book(name, author, price, numPages); // Create book object
        }

        // Display details of all books
        System.out.println("\nDetails of all books:");
        for (int i = 0; i < n; i++) {

```

```
        System.out.println("\nBook " + (i + 1) + ":\n" + books[i].toStringMethod());
    }
    scanner.close();
}
}
```

```
C:\Users\Aishwarya Vijay\Desktop\1BM22CS025>javac Books.java
```

```
C:\Users\Aishwarya Vijay\Desktop\1BM22CS025>java Books
```

```
Enter the number of books: 4
```

```
Enter details for Book 1:
```

```
Name: The Secret
```

```
Author: Rhonda Byrne
```

```
Price: $40
```

```
Number of Pages: 400
```

```
Enter details for Book 2:
```

```
Name: The Upside of falling
```

```
Author: Alex Light
```

```
Price: $35
```

```
Number of Pages: 350
```

```
Enter details for Book 3:
```

```
Name: The Summer Madness
```

```
Author: Marcus Vogel
```

```
Price: $34
```

```
Number of Pages: 350
```

```
Enter details for Book 4:
```

```
Name: Before the coffee gets cold
```

```
Author: Toshikazu Kawaguchi
```

```
Price: $90
```

```
Number of Pages: 500
```

Details of all books:

Book 1:

Book Details:

Name: The Secret, Author: Rhonda Byrne, Price: \$40.0, Number of Pages: 400

Book 2:

Book Details:

Name: The Upside of falling, Author: Alex Light, Price: \$35.0, Number of Pages: 350

Book 3:

Book Details:

Name: The Summer Madness, Author: Marcus Vogel, Price: \$34.0, Number of Pages: 350

Book 4:

Book Details:

Name: Before the coffee gets cold, Author: Toshikazu Kawaguchi, Price: \$90.0, Number of Pages: 500

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.

```
import java.util.Scanner;

abstract class Shape {
    protected int dimension1;
    protected int dimension2;

    // Constructor
    public Shape(int dimension1, int dimension2) {
        this.dimension1 = dimension1;
        this.dimension2 = dimension2;
    }

    // Abstract method to print area
    abstract void printArea();
}

class Rectangle extends Shape {
    public Rectangle(int length, int breadth) {
        super(length, breadth);
    }

    // Method to calculate and print area of rectangle
    void printArea() {
        System.out.println("Area of Rectangle: " + (dimension1 * dimension2));
    }
}

class Triangle extends Shape {
    public Triangle(int base, int height) {
        super(base, height);
    }
}
```

```

    }

    // Method to calculate and print area of triangle
    void printArea() {
        System.out.println("Area of Triangle: " + (0.5 * dimension1 * dimension2));
    }
}

class Circle extends Shape {
    public Circle(int radius) {
        super(radius, 0);
    }

    // Method to calculate and print area of circle
    void printArea() {
        System.out.println("Area of Circle: " + (Math.PI * dimension1 * dimension1));
    }
}

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

        // Taking dimensions from user
        System.out.print("Enter length and breadth of Rectangle: ");
        int length = scanner.nextInt();
        int breadth = scanner.nextInt();
        Rectangle rectangle = new Rectangle(length, breadth);

        System.out.print("Enter base and height of Triangle: ");
        int base = scanner.nextInt();
        int height = scanner.nextInt();
        Triangle triangle = new Triangle(base, height);

        System.out.print("Enter radius of Circle: ");
        int radius = scanner.nextInt();
        Circle circle = new Circle(radius);

        // Printing areas

```

```
    rectangle.printArea();  
    triangle.printArea();  
    circle.printArea();  
    scanner.close();  
}  
}
```

```
C:\Users\Aishwarya Vijay\Desktop\1BM22CS025>javac Dimensions.java
```

```
C:\Users\Aishwarya Vijay\Desktop\1BM22CS025>java Dimensions
```

```
Enter length and breadth of Rectangle: 12 6
```

```
Enter base and height of Triangle: 7 13
```

```
Enter radius of Circle: 4
```

```
Area of Rectangle: 72
```

```
Area of Triangle: 45.5
```

```
Area of Circle: 50.26548245743669
```

```
C:\Users\Aishwarya Vijay\Desktop\1BM22CS025>javac Dimensions.java
```

```
C:\Users\Aishwarya Vijay\Desktop\1BM22CS025>java Dimensions
```

```
Enter length and breadth of Rectangle: 44 7
```

```
Enter base and height of Triangle: 99 8
```

```
Enter radius of Circle: 44
```

```
Area of Rectangle: 308
```

```
Area of Triangle: 396.0
```

```
Area of Circle: 6082.123377349839
```

```
C:\Users\Aishwarya Vijay\Desktop\1BM22CS025>
```

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;

    long accountNumber;

    String accountType;

    double balance;

    // Constructor

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

        this.customerName = customerName;

        this.accountNumber = accountNumber;

        this.accountType = accountType;

        this.balance = 0.0;

    }

    // Method to accept deposit and update balance

    void deposit(double amount) {

        balance += amount;

        System.out.println("Deposit successful.");

    }

    // Method to display balance
```

```

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

// Method to compute and deposit interest
void depositInterest(double rate) {
    double interest = balance * (rate / 100);
    balance += interest;
    System.out.println("Interest deposited: $" + interest);
}

// Method to permit withdrawal and update balance
void withdraw(double amount) {
    if (balance >= amount) {
        balance -= amount;
        System.out.println("Withdrawal successful.");
    } else {
        System.out.println("Insufficient balance.");
    }
}
}

class CurAcct extends Account {
    double minBalance;
    double serviceCharge;

    // Constructor
    public CurAcct(String customerName, long accountNumber) {
        super(customerName, accountNumber, "Current");
        this.minBalance = 1000.0;
        this.serviceCharge = 50.0;
    }
}

```



```

// Method to check and impose penalty for minimum balance
void checkMinBalance() {
    if (balance < minBalance) {
        balance -= serviceCharge;
        System.out.println("Minimum balance charge imposed: $" + serviceCharge);
    }
}

}

class SavAcct extends Account {
    double interestRate;

    // Constructor
    public SavAcct(String customerName, long accountNumber) {
        super(customerName, accountNumber, "Savings");
        this.interestRate = 5.0; // 5% interest rate
    }

    // Method to apply compound interest
    void applyInterest() {
        depositInterest(interestRate);
    }
}

public class Bank {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Welcome to the Bank");

        // Creating a savings account
        SavAcct savingsAccount = new SavAcct("John Doe", 123456789);

```

```
// Creating a current account
```

```
CurAcct currentAccount = new CurAcct("Jane Doe", 987654321);
```

```
int choice;
```

```
do {
```

```
    System.out.println("\nMenu:");
```

```
    System.out.println("1. Deposit");
```

```
    System.out.println("2. Display Balance");
```

```
    System.out.println("3. Deposit Interest (Savings)");
```

```
    System.out.println("4. Withdraw");
```

```
    System.out.println("5. Exit");
```

```
    System.out.print("Enter your choice: ");
```

```
    choice = scanner.nextInt();
```

```
switch (choice) {
```

```
    case 1:
```

```
        System.out.print("Enter amount to deposit: ");
```

```
        double depositAmount = scanner.nextDouble();
```

```
        System.out.print("Enter account type (S for Savings, C for Current): ");
```

```
        char accountType = scanner.next().charAt(0);
```

```
        if (accountType == 'S') {
```

```
            savingsAccount.deposit(depositAmount);
```

```
        } else if (accountType == 'C') {
```

```
            currentAccount.deposit(depositAmount);
```

```
        } else {
```

```
            System.out.println("Invalid account type.");
```

```
        }
```

```
        break;
```

```
    case 2:
```

```
        System.out.print("Enter account type (S for Savings, C for Current): ");
```

```
        char accType = scanner.next().charAt(0);
```

```
        if (accType == 'S') {
```

```

        savingsAccount.displayBalance();
    } else if (accType == 'C') {
        currentAccount.displayBalance();
    } else {
        System.out.println("Invalid account type.");
    }
    break;
case 3:
    savingsAccount.applyInterest();
    System.out.println("Interest deposited successfully.");
    break;
case 4:
    System.out.print("Enter amount to withdraw: ");
    double withdrawAmount = scanner.nextDouble();
    System.out.print("Enter account type (S for Savings, C for Current): ");
    char type = scanner.next().charAt(0);
    if (type == 'S') {
        savingsAccount.withdraw(withdrawAmount);
    } else if (type == 'C') {
        currentAccount.withdraw(withdrawAmount);
        currentAccount.checkMinBalance();
    } else {
        System.out.println("Invalid account type.");
    }
    break;
case 5:
    System.out.println("Exiting...");
    break;
default:
    System.out.println("Invalid choice. Please try again.");
}
} while (choice != 5);

```

```
        scanner.close();  
    }  
}
```

```
Menu:  
1. Deposit  
2. Display Balance  
3. Deposit Interest (Savings)  
4. Withdraw  
5. Exit  
Enter your choice: 1  
Enter amount to deposit: 400000  
Enter account type (S for Savings, C for Current): S  
Deposit successful.
```

```
Menu:  
1. Deposit  
2. Display Balance  
3. Deposit Interest (Savings)  
4. Withdraw  
5. Exit  
Enter your choice: 4  
Enter amount to withdraw: 5000  
Enter account type (S for Savings, C for Current): S  
Withdrawal successful.
```

```
Menu:  
1. Deposit  
2. Display Balance  
3. Deposit Interest (Savings)  
4. Withdraw  
5. Exit  
Enter your choice: 2  
Enter account type (S for Savings, C for Current): S  
Balance: $395000.0
```

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.

// File: Student.java (inside CIE package)

```
package CIE;
```

```
public class Student {  
    String usn;  
    String name;  
    int sem;  
    // Other relevant members and methods...  
}
```

// File: Internals.java (inside CIE package)

```
package CIE;
```

```
public class Internals {  
    int[] internalMarks = new int[5]; // Array to store internal marks  
    // Other relevant members and methods...  
}
```

// File: External.java (inside SEE package)

```
package SEE;
```

```
import CIE.Student; // Import Student class from CIE package
```

```
public class External extends Student {
```

```
int[] seeMarks = new int[5]; // Array to store SEE marks
// Other relevant members and methods...
}

// File: Main.java (where you calculate final marks for n students)
import CIE.Student;
import SEE.External;

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

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;
```

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

```
class Father {  
    private int age;  
  
    public Father(int age) throws WrongAgeException {  
        if (age < 0) {  
            throw new WrongAgeException("Father's age cannot be negative.");  
        }  
        this.age = age;  
    }  
  
    public int getAge() {  
        return age;  
    }  
}
```

```
class Son extends Father {
```

```

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

    if (sonAge >= fatherAge) {
        throw new WrongAgeException("Son's age must be less than father's age.");
    }
}

}

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

        try {
            System.out.print("Enter father's age: ");
            int fatherAge = scanner.nextInt();

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

            Son son = new Son(fatherAge, sonAge);

            System.out.println("Father's age: " + fatherAge);
            System.out.println("Son's age: " + sonAge);
        } catch (WrongAgeException e) {
            System.out.println("Exception: " + e.getMessage());
        }
    }
}

```


Microsoft Windows [Version 10.0.22621.3155]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Aishwarya Vijay\Desktop\1BM22CS025>javac FatherSon.java

C:\Users\Aishwarya Vijay\Desktop\1BM22CS025>java FatherSon

Enter father's age: 56

Enter son's age: 23

Father's age: 56

Son's age: 23

C:\Users\Aishwarya Vijay\Desktop\1BM22CS025>javac FatherSon.java

C:\Users\Aishwarya Vijay\Desktop\1BM22CS025>java FatherSon

Enter father's age:

23

Enter son's age: 65

Exception: Son's age must be less than father's age.

C:\Users\Aishwarya Vijay\Desktop\1BM22CS025>|

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.

```
public class DisplayThreads {  
    public static void main(String[] args) {  
        Thread bmsThread = new Thread(() -> {  
            while (true) {  
                System.out.println("BMS College of Engineering");  
                try {  
                    Thread.sleep(10000); // Sleep for 10 seconds  
                } catch (InterruptedException e) {  
                    e.printStackTrace();  
                }  
            }  
        });  
  
        Thread cseThread = new Thread(() -> {  
            while (true) {  
                System.out.println("CSE");  
                try {  
                    Thread.sleep(2000); // Sleep for 2 seconds  
                } catch (InterruptedException e) {  
                    e.printStackTrace();  
                }  
            }  
        });  
  
        bmsThread.start();  
        cseThread.start();  
    }  
}
```

```
C:\Users\Aishwarya Vijay\Desktop\1BM22CS025>javac DisplayThreads.java
```

```
C:\Users\Aishwarya Vijay\Desktop\1BM22CS025>java DisplayThreads
```

```
CSE
```

```
BMS College of Engineering
```

```
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
```

