B.M.S. COLLEGE OF ENGINEERING

(Autonomous Institute, Affiliated to VTU)
Bull Temple Road, Basavanagudi, Bengaluru - 560019

Object Oriented Java Programing Alternate Assessment Tool (AAT)

Submitted in partial fulfilment of the requirements for the award of degree

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

By

GARGI BHARADWAJ **1BM22CS099**

Under the guidance of

Prof. Shravya
Assistant Professor

Department of Computer Science and Engineering
2023-2024

B.M.S. COLLEGE OF ENGINEERING

(Autonomous Institute, Affiliated to VTU)
Bull Temple Road, Basavanagudi,
Bengaluru – 560019

Department of Computer Science and Engineering

C E R T I F I C A T E

This is to certify that Alternate Assessment Tool is a bona-fide work carried out by **Gargi Bharadwaj(1BM22CS099)** in partial fulfilment for the award of degree of Bachelor of Engineering in **Computer Science and Engineering** from **Visvesvaraya Technological University, Belgaum** during the year **2023-2024**. It is certified that all corrections/suggestions indicated for Internal Assessments have been incorporated in the report deposited in the departmental library.

Prof. Shravya AR
Course Coordinator

Dr. Jyothi S Nayak
Professor and HOD

TABLE OF CONTENTS

LAB PROGRAMS

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, displaying a message stating that there are no real solutions.

CODE

```
import java.util.Scanner;

public class QuadraticEquationSolver {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
      System.out.println("Gargi Bharadwaj 1BM22CS099");
      System.out.println("Enter the coefficients of the quadratic equation (a, b, c):");
      double a = scanner.nextDouble();
      double b = scanner.nextDouble();
      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("Root 1: " + root1);
         System.out.println("Root 2: " + root2);
      } else if (discriminant == 0) {
         double root = -b / (2 * a);
         System.out.println("Root: " + root);
      } else {
         System.out.println("No real solutions exist.");
      }

      scanner.close();
   }
}
```

OUTPUT

```
java -cp /tmp/oXcuAHyhwS QuadraticEquationSolver
Gargi Bharadwaj 1BM22CS099
Enter the coefficients of the quadratic equation (a, b, c):
1
2
1
Root: -1.0
```

```
java -cp /tmp/oXcuAHyhwS QuadraticEquationSolver
Gargi Bharadwaj 1BM22CS099
Enter the coefficients of the quadratic equation (a, b, c):
3
64

4
No real solutions exist.
```

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.

CODE

```
import java.util.Scanner;

public class StudentMain {
    public static void main(String[] args) {
        Student student = new Student();
        System.out.println("GARGI BHARADWAJ 1BM22CS099");
        student.acceptDetails();
        student.displayDetails();
        System.out.println("SGPA: " + student.calculateSGPA());
    }
}
class Student {
    private String usn;
    private String name;
    private int[] credits;
    private int[] marks;

    public void acceptDetails() {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter USN:");
        usn = scanner.next();
        System.out.println("Enter Name:");
        name = scanner.next();
        System.out.println("Enter the number of subjects:");
        int numSubjects = scanner.nextInt();
        credits = new int[numSubjects];
        marks = new int[numSubjects];
        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();
        }
        scanner.close();
    }

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

```
        System.out.println("Credits:");
        for (int credit : credits) {
            System.out.print(credit + " ");
        }
        System.out.println("\nMarks:");
        for (int mark : marks) {
            System.out.print(mark + " ");
        }
        System.out.println();
    }

    public float calculateSGPA() {
        int totalCredits = 0;
        float totalPoints = 0;
        for (int i = 0; i < credits.length; i++) {
            totalCredits += credits[i];
            if (marks[i] >= 90)
                totalPoints += 10 * credits[i];
            else if (marks[i] >= 80)
                totalPoints += 9 * credits[i];
            else if (marks[i] >= 70)
                totalPoints += 8 * credits[i];
            else if (marks[i] >= 60)
                totalPoints += 7 * credits[i];
            else if (marks[i] >= 50)
                totalPoints += 6 * credits[i];
            else if (marks[i] >= 40)
                totalPoints += 5 * credits[i];
        }
        return totalPoints / totalCredits;
    }
}
```

OUTPUT

```
java -cp /tmp/oXcuAHyhwS StudentMain
GARGI BHARADWAJ 1BM22CS099
Enter USN:
1
Enter Name:
GARGI
Enter the number of subjects:
2
Enter credits for subject 1:
9
Enter marks for subject 1:
99
Enter credits for subject 2:
5
Enter marks for subject 2:
98
USN: 1
Name: GARGI
Credits:
9 5
Marks:
99 98
SGPA: 10.0
```

LAB PROGRAM 3

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

CODE

```
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("GARGI BHARADWAJ 1BM22CS099 ");
        System.out.print("Enter the number of books to create: ");
        int n = scanner.nextInt();

        Book[] books = new Book[n];

        for (int i = 0; i < n; i++) {
            System.out.println("Enter details for Book " + (i + 1) + ":");
            System.out.print("Name: ");
            String name = scanner.next();
            System.out.print("Author: ");
            String author = scanner.next();
            System.out.print("Price: ");
            double price = scanner.nextDouble();
            System.out.print("Number of Pages: ");
            int numPages = scanner.nextInt();

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

        System.out.println("\nDetails of the books:");
        for (int i = 0; i < n; i++) {
            System.out.println("Book " + (i + 1) + ":");
            System.out.println(books[i]);
            System.out.println();
        }

        scanner.close();
    }
}
class Book {
    private String name;
    private String author;
    private double price;
```

```
    private int numPages;

    public Book(String name, String author, double price, int numPages) {
        this.name = name;
        this.author = author;
        this.price = price;
        this.numPages = numPages;
    }

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

    public void setAuthor(String author) {
        this.author = author;
    }

    public void setPrice(double price) {
        this.price = price;
    }

    public void setNumPages(int numPages) {
        this.numPages = numPages;
    }

    public String getName() {
        return name;
    }

    public String getAuthor() {
        return author;
    }

    public double getPrice() {
        return price;
    }

    public int getNumPages() {
        return numPages;
    }

    public String toString() {
            return "Book Details:\nName: " + name + "\nAuthor: " + author + "\nPrice: $" + price +
"\nNumber of Pages: " + numPages;
    }
```

}

OUTPUT

```
java -cp /tmp/oXcuAHyhwS Main
GARGI BHARADWAJ 1BM22CS099
Enter the number of books to create: 2
Enter details for Book 1:
Name: Joy
Author: JP
Price: 300
Number of Pages: 400
Enter details for Book 2:
Name: Giddy
Author: Goldy
Price: 250
Number of Pages: 421

Details of the books:
Book 1:
Book Details:
Name: Joy
Author: JP
Price: $300.0
Number of Pages: 400

Book 2:
Book Details:
Name: Giddy
Author: Goldy
Price: $250.0
Number of Pages: 421
```

LAB 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.

CODE

```
abstract class Shape {
    protected int side1;
    protected int side2;

    public Shape(int side1, int side2) {
        this.side1 = side1;
        this.side2 = side2;
    }

    public abstract void printArea();
}

class Rectangle extends Shape {
    public Rectangle(int side1, int side2) {
        super(side1, side2);
    }

    public void printArea() {
        int area = side1 * side2;
        System.out.println("Area of Rectangle: " + area);
    }
}

class Triangle extends Shape {
    public Triangle(int side1, int side2) {
        super(side1, side2);
    }

    public void printArea() {
        double area = 0.5 * side1 * side2;
        System.out.println("Area of Triangle: " + area);
    }
}

class Circle extends Shape {
    public Circle(int side1, int side2) {
        super(side1, side2);
```

```
    }

    public void printArea() {
        double area = Math.PI * side1 * side1;
        System.out.println("Area of Circle: " + area);
    }
}

public class Main {
    public static void main(String[] args) {
        System.out.println("GARGI BHARADWAJ 1BM22CS099");
        Rectangle rectangle = new Rectangle(5, 7);
        rectangle.printArea();

        Triangle triangle = new Triangle(4, 9);
        triangle.printArea();

        Circle circle = new Circle(6, 0); // only one side required for circle
        circle.printArea();
    }
}
```

OUTPUT

```
java -cp /tmp/ZVk0FnZHdc Main
GARGI BHARADWAJ 1BM22CS099
Area of Rectangle: 35
Area of Triangle: 18.0
Area of Circle: 113.09733552923255
```

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.

CODE

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

      System.out.println("Enter savings account details:");
      System.out.print("Customer Name: ");
      String savingsCustomerName = scanner.nextLine();
      System.out.print("Account Number: ");
      int savingsAccountNumber = scanner.nextInt();
      System.out.print("Initial Balance: ");
      double savingsBalance = scanner.nextDouble();
      System.out.print("Interest Rate: ");
      double savingsInterestRate = scanner.nextDouble();
            SavingsAccount savingsAccount = new SavingsAccount(savingsCustomerName,
savingsAccountNumber, savingsBalance, savingsInterestRate);

      System.out.println("Enter current account details:");
      scanner.nextLine();
      System.out.print("Customer Name: ");
      String currentCustomerName = scanner.nextLine();
      System.out.print("Account Number: ");
      int currentAccountNumber = scanner.nextInt();
      System.out.print("Initial Balance: ");
      double currentBalance = scanner.nextDouble();
      System.out.print("Minimum Balance: ");
      double minimumBalance = scanner.nextDouble();
```

```
        System.out.print("Service Charge: ");
        double serviceCharge = scanner.nextDouble();
                CurrentAccount currentAccount = new CurrentAccount(currentCustomerName,
currentAccountNumber, currentBalance, minimumBalance, serviceCharge);

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

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

        System.out.println("Enter withdrawal amount for savings account:");
        double savingsWithdrawalAmount = scanner.nextDouble();
        savingsAccount.withdraw(savingsWithdrawalAmount);
        savingsAccount.displayBalance();

        System.out.println("Enter withdrawal amount for current account:");
        double currentWithdrawalAmount = scanner.nextDouble();
        currentAccount.withdraw(currentWithdrawalAmount);
        currentAccount.displayBalance();

        savingsAccount.depositInterest();

        System.out.println("Enter withdrawal amount for current account:");
        currentWithdrawalAmount = scanner.nextDouble();
        currentAccount.withdraw(currentWithdrawalAmount);
        currentAccount.displayBalance();

        scanner.close();
    }
}

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

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

```
        this.accountNumber = accountNumber;
        this.accountType = accountType;
        this.balance = balance;
    }

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

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

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

class SavingsAccount extends Account {
    private double interestRate;

        public SavingsAccount(String customerName, int accountNumber, double balance, double
interestRate) {
        super(customerName, accountNumber, "Savings", balance);
        this.interestRate = interestRate;
    }

    public void depositInterest() {
        double interest = balance * interestRate / 100;
        balance += interest;
        System.out.println("Interest deposited. New balance: " + balance);
    }
}

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

```
    public CurrentAccount(String customerName, int accountNumber, double balance, double minimumBalance, double serviceCharge) {
        super(customerName, accountNumber, "Current", balance);
        this.minimumBalance = minimumBalance;
        this.serviceCharge = serviceCharge;
    }

    @Override
    public void withdraw(double amount) {
        if (balance - amount >= minimumBalance) {
            balance -= amount;
            System.out.println("Withdrawal successful. New balance: " + balance);
        } else {
            System.out.println("Insufficient funds! Service charge of " + serviceCharge + " applied.");
            balance -= serviceCharge;
        }
    }
}
```

OUTPUT

```
java -cp /tmp/qbUCOd2RCn Main
Enter savings account details:
Customer Name: Gargi Bharadwaj
Account Number: 12344
Initial Balance: 6000
Interest Rate: 7
Enter current account details:
Customer Name: Gargi Bharadwaj
Account Number: 12344
Initial Balance: 5000
Minimum Balance: 2000
Service Charge: 4
Enter deposit amount for savings account:
3000
Deposit successful. New balance: 9000.0
Account Balance: 9000.0
Enter deposit amount for current account:
1000
Deposit successful. New balance: 6000.0
Account Balance: 6000.0
Enter withdrawal amount for savings account:
4000
Withdrawal successful. New balance: 5000.0
Account Balance: 5000.0
Enter withdrawal amount for current account:
2500
Withdrawal successful. New balance: 3500.0
Account Balance: 3500.0
Interest deposited. New balance: 5350.0
Enter withdrawal amount for current account:
```

LAB PROGRAM 6

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

CODE

```
import CIE.Student;
import CIE.Internals;
import SEE.External;

public class Main {
    public static void main(String[] args) {
        System.out.println("GARGI BHARADWAJ 1BM22CS099");
        int n = 3;
        Student[] students = new Student[n];
        students[0] = new Student("1", "Alice", 3);
        students[1] = new Student("2", "Bob", 3);
        students[2] = new Student("3", "Charlie", 3);
        Internals[] internals = new Internals[n];
        internals[0] = new Internals(new int[]{80, 75, 90, 85, 95});
        internals[1] = new Internals(new int[]{70, 85, 80, 90, 75});
        internals[2] = new Internals(new int[]{85, 90, 95, 80, 70});
        External[] externals = new External[n];
        externals[0] = new External("1", "Alice", 3, new int[]{75, 80, 85, 90, 95});
        externals[1] = new External("2", "Bob", 3, new int[]{80, 85, 90, 95, 70});
        externals[2] = new External("3", "Charlie", 3, new int[]{85, 90, 95, 70, 80});
        for (int i = 0; i < n; i++) {
            System.out.println("Student: " + students[i].name);
            System.out.println("USN: " + students[i].USN);
            System.out.println("Semester: " + students[i].sem);
            System.out.println("Final Marks:");
            for (int j = 0; j < 5; j++) {
                int finalMarks = internals[i].marks[j] + externals[i].marks[j];
                System.out.println("Course " + (j + 1) + ": " + finalMarks);
            }
            System.out.println();
        }
    }
}

package CIE;
```

```
public class Student {
    protected String USN;
    protected String name;
    protected int sem;

    public Student(String USN, String name, int sem) {
        this.USN = USN;
        this.name = name;
        this.sem = sem;
    }
}

public class Internals {
    protected int[] marks = new int[5];

    public Internals(int[] marks) {
        this.marks = marks;
    }
}
package SEE;
import CIE.Student;

public class External extends Student {
    protected int[] marks = new int[5];

    public External(String USN, String name, int sem, int[] marks) {
        super(USN, name, sem);
        this.marks = marks;
    }
}
```

OUTPUT

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.

CODE

```
import java.util.*;
public class Main {
   public static void main(String[] args) {
      Scanner sc=new Scanner(System.in);
      System.out.println("GARGI BHARADWAJ 1BM22CS099");
      try {
         System.out.println("Father's age:");
         int n=sc.nextInt();
         System.out.println("Son's age:");
         int m=sc.nextInt();
         Son son = new Son(n,m);
         Father father = new Father(n);
         System.out.println("Father's Age: " + father.getAge());

         System.out.println("Son's Age: " + son.getSonAge());
      } catch (WrongAge e) {
         System.out.println("Error: " + e.getMessage());
      }
   }
}

class WrongAge extends Exception {
   public WrongAge() {
      super("Invalid age provided.");
   }
}

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);
        if (sonAge >= fatherAge) {
            throw new WrongAge();
        }
        this.sonAge = sonAge;
    }

    public int getSonAge() {
        return sonAge;
    }
}
```

OUTPUT

```
GARGI BHARADWAJ 1BM22CS099
Father's age:
12
Son's age:
34
ERROR!
Error: Invalid age provided.
```

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.

CODE

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

class DisplayCSE extends Thread {
   public void run() {
      while (true) {
         System.out.println("CSE");
         try {
            Thread.sleep(2000);
         } catch (InterruptedException e) {
            e.printStackTrace();
         }
      }
   }
}

public class Main {
   public static void main(String[] args) {
      DisplayBMSCE bmsceThread = new DisplayBMSCE();
      DisplayCSE cseThread = new DisplayCSE();

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

OUTPUT

```
Gargi Bharadwaj 1BM22CS099
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
```