B.M.S. College of Engineering

(Autonomous Institution affiliated to VTU, Belagavi)

Department of Computer Science and Engineering

LAB

OBJECT ORIENTED JAVA PROGRAMMING REPORT

23CS3PCOOJ

(December 2023-March 2024)

B.M.S. College of Engineering
Department of Computer Science and Engineering

Laboratory Certificate

This is to certify that BHARATH C has satisfactorily completed the course of Experiments in Practical OBJECT-ORIENTED JAVA PROGRAMMING prescribed by the Department during the odd semester 2023-24.
Name of the Candidate: BHARATH C

USN No: **1BM22CS068** Semester: **III** Section: **B**

Marks	
Max. Marks	Obtained
10	
Marks in Words	

Signature of the staff in-charge **Head of the Department**

INDEX

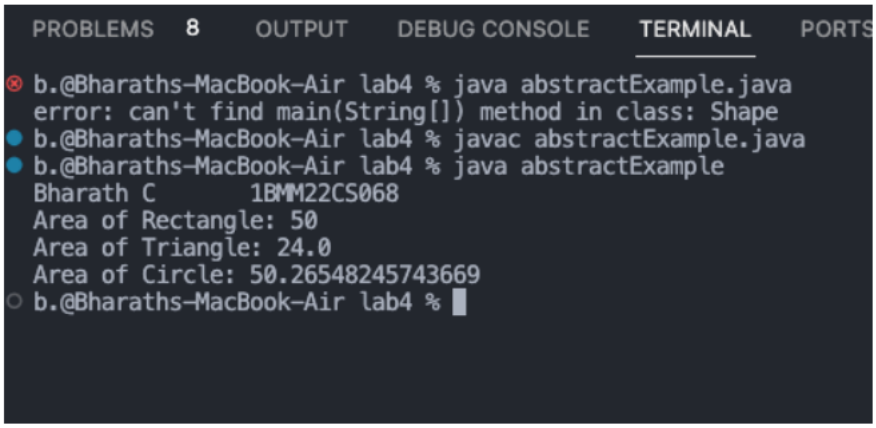
Pgm-1: Program to find quadratic equation of given roots.

Solution:

```
import java.util.Scanner;

class quadratic {
    int a, b, c;
    double r1, r2, d;

    void getData() {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter the values of a, b, c");
        a = s.nextInt();
        b = s.nextInt();
        c = s.nextInt();
    }

    void compute() {
        while (a == 0) {
            System.out.println("Not a quadratic equation");
            System.out.println("Enter a non-zero value of a");
            Scanner s = new Scanner(System.in);
            a = s.nextInt();
        }

        d = (b * b) - (4 * a * c);

        if (d == 0) {
            r1 = -b / (2 * (double) a);
            System.out.println("Roots are real and equal");
            System.out.println("Roots are Root1=Root2=" + r1);
        } else if (d > 0) {
            r1 = (-b + Math.sqrt(d)) / (2 * (double) a);
            r2 = (-b - Math.sqrt(d)) / (2 * (double) a);
            System.out.println("Roots are real and distinct");
            System.out.println("Roots are Root1=" + r1 + " and Root2=" + r2);
        } else {
            r1 = -b / (2 * (double) a);
            r2 = Math.sqrt(Math.abs(d)) / (2 * (double) a);
            System.out.println("Roots are imaginary and Root1=" + r1 + "+i" + r2 +
" and Root2=" + r1 + "-i" + r2);
        }
    }
}

class quadraticMain {
    public static void main(String[] args) {
        System.out.println("Bharath C\t1BMM22CS068");
```

```
        quadratic q = new quadratic();
        q.getData();
        q.compute();
    }
}
```

Output:



Pgm-2: Create a class Book that contains four members: name, author, price, and 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.

Solution:

```
import java.util.Scanner;

class book {
    String name;
    String author;
    int price;
    int numPages;

    book(String name, String author, int price, int numPages) {
        this.name = name;
        this.author = author;
        this.price = price;
        this.numPages = numPages;
    }

    public String toString() {
        return "Book Name: " + this.name + "\n" +
                "Author Name: " + this.author + "\n" +
                "Book Price: " + this.price + "\n" +
                "Number of pages: " + this.numPages + "\n";
    }
}

class bookMain {
    public static void main(String[] args) {
        System.out.println("Bharath C\t1BMM22CS068");
        Scanner s = new Scanner(System.in);
        int n;
        String name;
        String author;
        int price;
        int numPages;

        System.out.println("Enter the number of book:");
        n = s.nextInt();

        book[] b;
        b = new book[n];

        for (int i = 0; i < n; i++) {
            System.out.println("Book " + (i + 1) + ":");
```

```
            System.out.println("Enter the book name");
            s.nextLine();
            name = s.nextLine();
            System.out.println("Enter the author");
            author = s.nextLine();
            System.out.println("Enter the price");
            price = s.nextInt();
            System.out.println("Enter the number of pages");
            numPages = s.nextInt();

            b[i] = new book(name, author, price, numPages);
        }

        for (int i = 0; i < n; i++) {
            System.out.println("Book " + (i + 1) + "\n" + b[i]);
        }
    }
}
```

Output:

```
b.@Bharaths-MacBook-Air lab3 % javac book.java
b.@Bharaths-MacBook-Air lab3 % java bookMain
Bharath C       1BMM22CS068
Enter the number of book:
2
Book 1:
Enter the book name
Davinci Code
Enter the author
Dan brown
Enter the price
700
Enter the number of pages
890
Book 2:
Enter the book name
Let it Snow
Enter the author
John Greene
Enter the price
400
Enter the number of pages
300
Book 1
Book Name: Davinci Code
Author Name: Dan brown
Book Price: 700
Number of pages: 890

Book 2
Book Name: Let it Snow
Author Name: John Greene
Book Price: 400
Number of pages: 300
```

Pgm-3: Write a Java program to create a class Student with members USN, name, marks (6 subjects). Include methods to accept student details and marks, also include a method to calculate the percentage and display appropriate details. (Array of student object to be created).

Solution:

```
import java.util.Scanner;

class student {
    String USN;
    String name;
    int marks[] = new int[6];
    float percentage = 0;

    void getData(int i) {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter USN: ");
        USN = s.next();
        System.out.println("Enter Name:");
        name = s.next();
        System.out.println("Enter Student" + i + " Marks");
        for (int j = 0; j < 6; j++) {
            System.out.println("Enter Marks of Subject" + j + ":");
            marks[j] = s.nextInt();
            percentage += marks[j];
        }
    }

    void calculatePercentage(int i) {
        percentage = (percentage / 6);
        System.out.println("Percentage of student" + i + "=" + percentage + "%");
    }
}

class studentMain {
    public static void main(String[] args) {
      System.out.println("Bharath C\t1BMM22CS068");
        System.out.println("Enter the number of Students");
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        student s[] = new student[n];
        for (int i = 0; i < n; i++) {
            s[i] = new student();
            s[i].getData(i);
```

```
        }
        for (int i = 0; i < n; i++) {
            s[i].calculatePercentage(i);
        }
    }
}
```

Output:

```
b.@Bharaths-MacBook-Air lab3 % javac student.java
b.@Bharaths-MacBook-Air lab3 % java run
Bharath C       1BMM22CS068
Enter the number of Students
2
Enter USN:
001
Enter Name:
Bharath
Enter Student0 Marks
Enter Marks of Subject0:
90
Enter Marks of Subject1:
92
Enter Marks of Subject2:
89
Enter Marks of Subject3:
100
Enter Marks of Subject4:
98
Enter Marks of Subject5:
95
Enter USN:
002
Enter Name:
Raj
Enter Student1 Marks
Enter Marks of Subject0:
89
Enter Marks of Subject1:
98
Enter Marks of Subject2:
89
Enter Marks of Subject3:
98
Enter Marks of Subject4:
100
Enter Marks of Subject5:
99
Percentage of student0=94.0%
Percentage of student1=95.5%
b.@Bharaths-MacBook-Air lab3 %
```

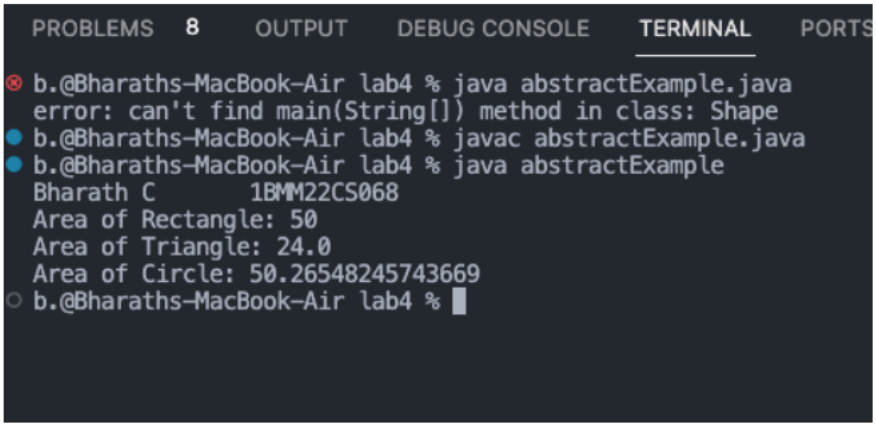
Pgm-4: Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea( ). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contain the method printArea( ) that prints the area of the given shape.

Solution:

```
abstract class Shape {
    // Two integers representing dimensions
    int dimension1;
    int dimension2;

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

    // Abstract method to be implemented by subclasses
    abstract void printArea();
}

class Rectangle extends Shape {
    // Constructor
    Rectangle(int length, int width) {
        super(length, width);
    }

    // Implementation of printArea for Rectangle
    @Override
    void printArea() {
        int area = dimension1 * dimension2;
        System.out.println("Area of Rectangle: " + area);
    }
}

class Triangle extends Shape {
    // Constructor
    Triangle(int base, int height) {
        super(base, height);
    }

    // Implementation of printArea for Triangle
    @Override
    public void printArea() {
        double area = 0.5 * dimension1 * dimension2;
        System.out.println("Area of Triangle: " + area);
    }
```

```java
}

class Circle extends Shape {
    // Constructor
    Circle(int radius) {
        super(radius, 0); // Only one dimension needed for a circle
    }

    // Implementation of printArea for Circle
    @Override
    void printArea() {
        double area = Math.PI * dimension1 * dimension1;
        System.out.println("Area of Circle: " + area);
    }
}

class abstractExample {
    public static void main(String[] args) {
        // Creating objects of each shape
        System.out.println("Bharath C\t1BMM22CS068");
        Rectangle rectangle = new Rectangle(5, 10);
        Triangle triangle = new Triangle(8, 6);
        Circle circle = new Circle(4);

        // Printing areas
        rectangle.printArea();
        triangle.printArea();
        circle.printArea();
    }
}
```

Output:



Pgm-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:

- Accept deposit from customer and update the balance.
- Display the balance.
- Compute and deposit interest
- Permit withdrawal and update the balance

Check for the minimum balance, impose penalty if necessary and update the balance.

Solution:

```
import java.util.Scanner;

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

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

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

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

```
    }

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

    void computeInterest() {
    }

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

class SavAcct extends Account {
    double interestRate = 0.05;

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

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

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

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

    void checkMinimumBalance(double minBalance, double serviceCharge) {
        if (balance < minBalance) {
            System.out.println("Service charge of INR " + serviceCharge + "
imposed");
            balance -= serviceCharge;
        }
    }
}
class bankDataBase {
    public static void main(String[] args) {
```

```
        try (Scanner scanner = new Scanner(System.in)) {
            System.out.println("Bharath C\t1BMM22CS068");
            System.out.print("Enter the number of users: ");
            int numUsers = scanner.nextInt();

            Account[] accounts = new Account[numUsers];

            for (int i = 0; i < numUsers; i++) {
                System.out.println("\nUser " + (i + 1));
                System.out.print("Enter customer name: ");
                scanner.nextLine();
                String name = scanner.nextLine();
                System.out.print("Enter account number: ");
                int accNumber = scanner.nextInt();
                System.out.print("Enter initial deposit amount: INR ");
                double initialDeposit = scanner.nextDouble();
                System.out.print("Enter account type (Savings/Current): ");
                scanner.nextLine();
                String accType = scanner.nextLine();

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

            boolean exit = false;
            while (!exit) {
                System.out.println("\nChoose an option:");
                System.out.println("1. Deposit");
                System.out.println("2. Withdraw");
                System.out.println("3. Display Balance");
                System.out.println("4. Compute Interest (Savings only)");
                System.out.println("5. Exit");
                System.out.print("Enter your choice: ");
                int choice = scanner.nextInt();

                switch (choice) {
                    case 1:
                        System.out.print("Enter account number: ");
                        int accNum = scanner.nextInt();
                        System.out.print("Enter deposit amount: INR ");
                        double depositAmount = scanner.nextDouble();
```

```
                        for (Account acc : accounts) {
                            if (acc.accountNumber == accNum) {
                                acc.deposit(depositAmount);
                            }
                        }
                        break;
                    case 2:
                        System.out.print("Enter account number: ");
                        accNum = scanner.nextInt();
                        System.out.print("Enter withdrawal amount: INR ");
                        double withdrawAmount = scanner.nextDouble();
                        for (Account acc : accounts) {
                            if (acc.accountNumber == accNum) {
                                acc.withdraw(withdrawAmount);
                            }
                        }
                        break;
                    case 3:
                        System.out.print("Enter account number: ");
                        accNum = scanner.nextInt();
                        for (Account acc : accounts) {
                            if (acc.accountNumber == accNum) {
                                acc.displayBalance();
                            }
                        }
                        break;
                    case 4:
                        System.out.print("Enter account number (for Savings
account): ");
                        accNum = scanner.nextInt();
                        for (Account acc : accounts) {
                            if (acc.accountNumber == accNum && acc instanceof
SavAcct) {
                                ((SavAcct) acc).computeInterest();
                            }
                        }
                        break;
                    case 5:
                        exit = true;
                        break;
                    default:
                        System.out.println("Invalid choice. Please enter a valid
option.");
                }
            }
        }

    }
}
```

Output:

```
b.@Bharaths-MacBook-Air lab4 % javac bankDataBase.java
b.@Bharaths-MacBook-Air lab4 % java bankDataBase
Bharath C       1BM22CS068
Enter the number of users: 2

User 1
Enter customer name: bharath
Enter account number: 001
Enter initial deposit amount: INR 1200
Enter account type (Savings/Current): Savings

User 2
Enter customer name: raj

Enter account number: 002
Enter initial deposit amount: INR 1300
Enter account type (Savings/Current): Current

Choose an option:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings only)
5. Exit
Enter your choice: 1
Enter account number: 001
Enter deposit amount: INR 1000
Deposit of INR 1000.0 successful

Choose an option:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings only)
5. Exit
Enter your choice: 3
Enter account number: 001
Account Number: 1
Customer Name: bharath
Account Type: Savings
Balance: INR 2200.0

Choose an option:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings only)
5. Exit
Enter your choice: 2
Enter account number: 3000
Enter withdrawal amount: INR 001

Choose an option:
1. Deposit
2. Withdraw
```

Pgm-6: Create a package CIE which has two classes- Student and Internals. The class Student has members like usn, name, sem. The class internals derived from student 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.

- Create a folder CIE and save the programs Student.java and Internals.java within it.
- Create a folder SEE and save the program External.java within it.
- Save the Main program outside these two folders.
- Compile Main.java and Execute the Main.class

Solution:

```java
// File: labPrograms/lab5/cie/internals.java
package labPrograms.lab5.cie;

public class internals extends student {
    public int[] internalMarks;

    public internals(String usn, String name, int sem, int[] marks) {
        super(usn, name, sem);
        this.internalMarks = marks;
    }
}
```

```java
// File: labPrograms/lab5/cie/student.java
package labPrograms.lab5.cie;

public class student {
    public String usn;
    public String name;
    public int sem;

    public student(String usnInp, String nameInp, int semInp) {
        this.usn = usnInp;
        this.name = nameInp;
        this.sem = semInp;
    }
}
```

```java
// File: labPrograms/lab5/see/external.java
package labPrograms.lab5.see;

import labPrograms.lab5.cie.student;

public class external extends student {
    public int[] seeMarks;

    public external(String usn, String name, int sem, int[] seeMarks) {
        super(usn, name, sem);
        this.seeMarks = seeMarks;
    }
}
```

```java
// File: labPrograms/lab5/see/see.java
package labPrograms.lab5.see;

// import labPrograms.lab5.see.external;
import java.util.Scanner;

import labPrograms.lab5.cie.internals;
//import labPrograms.lab5.see.external;

public class see {
    public static void main(String[] args) {
        System.out.println("Bharath C\t1BMM22CS068");

        try (Scanner scanner = new Scanner(System.in)) {
          System.out.print("Enter the number of students: ");
          int n = scanner.nextInt();

          internals[] cieStudents = new internals[n];
          external[] seeStudents = new external[n];

          // Input CIE marks
          for (int i = 0; i < n; i++) {
              System.out.println("Enter details for CIE of student " + (i + 1));
              System.out.print("USN: ");
              String usn = scanner.next();
              System.out.print("Name: ");
              String name = scanner.next();
              System.out.print("Semester: ");
              int sem = scanner.nextInt();

              int[] cieMarks = new int[5];
              System.out.print("Enter CIE marks for 5 courses: ");
              for (int j = 0; j < 5; j++) {
                  cieMarks[j] = scanner.nextInt();
              }
```

```
            cieStudents[i] = new internals(usn, name, sem, cieMarks); // Pass
cieMarks as an array
        }

        // Input SEE marks
        for (int i = 0; i < n; i++) {
            System.out.println("Enter details for SEE of student " + (i + 1));
            System.out.print("USN: ");
            String usn = scanner.next();
            System.out.print("Name: ");
            String name = scanner.next();
            System.out.print("Semester: ");
            int sem = scanner.nextInt();

            int[] seeMarks = new int[5];
            System.out.print("Enter SEE marks for 5 courses: ");
            for (int j = 0; j < 5; j++) {
                seeMarks[j] = scanner.nextInt();
            }

            seeStudents[i] = new external(usn, name, sem, seeMarks); // Pass
seeMarks as an array
        }

        // Displaying final marks
        System.out.println("\nFinal Marks of Students:");
        for (int i = 0; i < n; i++) {
            System.out.println("\nDetails of Student " + (i + 1));
            System.out.println("USN: " + cieStudents[i].usn);
            System.out.println("Name: " + cieStudents[i].name);
            System.out.println("Semester: " + cieStudents[i].sem);
            System.out.println("CIE Marks: ");
            for (int j = 0; j < 5; j++) {
                System.out.print(cieStudents[i].internalMarks[j] + " ");
            }
            System.out.println("\nSEE Marks: ");
            for (int j = 0; j < 5; j++) {
                System.out.print(seeStudents[i].seeMarks[j] + " ");
            }
        }
    }
}
```

Output:

```
b.@Bharaths-MacBook-Air 3-OOJ %  /usr/bin/env /Library/Java/
s/b./Library/Application\ Support/Code/User/workspaceStorage/
Search documents and filenames for text
Enter details for CIE of student 1
USN: 001
Name: b
Semester: 3
Enter CIE marks for 5 courses: 99 98 96 96 96
Enter details for CIE of student 2
USN: 002
Name: c
Semester: 3
Enter CIE marks for 5 courses: 88 87 86 85 85
Enter details for SEE of student 1
USN: 001
Name: b
Semester: 3
Enter SEE marks for 5 courses: 99 99 99 99 99
Enter details for SEE of student 2
USN: 002
Name: c
Semester: 3
Enter SEE marks for 5 courses: 88 88 88 88 88

Final Marks of Students:

Details of Student 1
USN: 001
Name: b
Semester: 3
CIE Marks:
99 98 96 96 96
SEE Marks:
99 99 99 99 99
Details of Student 2
USN: 002
Name: c
Semester: 3
CIE Marks:
88 87 86 85 85
SEE Marks:
88 88 88 88 88
b.@Bharaths-MacBook-Air 3-OOJ %
```

Pgm-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.

Solution:

```
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 {
    private int sonAge;

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

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

        this.sonAge = sonAge;
    }

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

```
}

class run {
    public static void main(String[] args) {
        System.out.println("Bharath C\t1BMM22CS068");
        try {
            Father father = new Father(40);
            System.out.println("Father's age: " + father.getAge());

            Son son = new Son(40, 20); // This will throw an exception
            System.out.println("Son's age: " + son.getSonAge());
        } catch (WrongAgeException e) {
            System.out.println("Error: " + e.getMessage()); // More specific error
message
    }
}
```

Output:

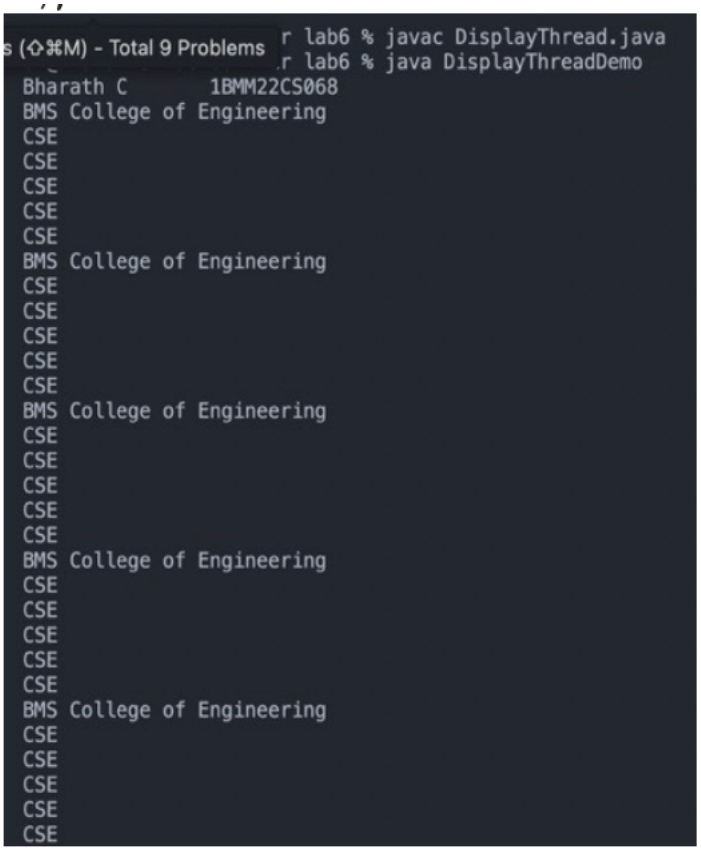
```
b.@Bharaths-MacBook-Air lab6 % javac WrongAgeException.java
b.@Bharaths-MacBook-Air lab6 % java run
Bharath C       1BMM22CS068
Father's age: 40
Son's age: 20
b.@Bharaths-MacBook-Air lab6 %
```

Pgm-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.

Solution:

```
class DisplayThread extends Thread {

     String message;
int intervalMillis;

   DisplayThread(String message, int intervalMillis) {
        this.message = message;
        this.intervalMillis = intervalMillis;
    }

    @Override // Recommended to explicitly override run() from Thread
    public void run() {
        while (true) {
            try {
                System.out.println(message);
                Thread.sleep(intervalMillis);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}

 class DisplayThreadDemo {

    public static void main(String[] args) {
        System.out.println("Bharath C\t1BMM22CS068");
        DisplayThread thread1 = new DisplayThread("BMS College of Engineering",
10000);
        DisplayThread thread2 = new DisplayThread("CSE", 2000);

        thread1.start();
        thread2.start();
    }
}
```

Output:


s (⇧⌘M) - Total 9 Problems
r lab6 % javac DisplayThread.java
r lab6 % java DisplayThreadDemo
Bharath C 1BMM22CS068
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
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE