OBJECT ORIENTED JAVA PROGRAMMING REPORT

A Report Submitted to

Visvesvaraya Technological University,
Belagavi

in partial fulfillment of requirements of
III semester OBJECT ORIENTED JAVA PROGRAMMING– 23CS3PCOOJ
of
BACHELOR OF ENGINEERING in COMPUTER SCIENCE ENGINEERING

Submitted by

Harsh Ticku **1BM22CS109**

Under the guidance of

SHRAVYA AR
Assistant Professor
Department of Computer Science Engineering

Department of Computer Science Engineering
B. M. S. COLLEGE OF ENGINEERING
(Autonomous Institution Affiliated to Visvesvaraya Technological University, Belagavi)
Post Box No. 1908, Bull Temple Road, Bengaluru –

560 019 August - 2023

B. M. S. COLLEGE OF ENGINEERING

Post Box No. 1908, Bull Temple Road, Bengaluru – 560 019

Department of Mechanical Engineering

Certificate

This is to certify that the work entitled "Experiments in Practical OBJECT ORIENTED JAVA PROGRAMMING" has been carried out by

Harsh Ticku
1BM22CS109

in partial fulfillment for the award of Bachelor of Engineering in Computer Science Engineering from Visvesvaraya Technological University, Belgaum, during the academic year 2023–24.

I hereby approve the report as it fulfills the academic requirements for the course **OBJECT ORIENTED JAVA PROGRAMMING (23CS3PCOOJ)** prescribed for the aforementioned degree.

Signature: Harsh Ticku
Date:25-02-2024

Marks	
Max.Marks	Obtained
10	

Signature of Guide
Shravya AR

Signature of HOD

Declaration

I hereby declare that the report entitled "**Experiments in Practical OBJECT ORIENTED JAVA PROGRAMMING**" has been independently carried out by me at the Department of Computer Science Engineering, under the guidance of Shravya AR, Assistant Professor, Department of Computer Science Engineering, B. M. S. College of Engineering, Bengaluru. This work is in partial fulfillment of the requirements for the degree of Bachelor of Engineering in Computer Science Engineering from Visvesvaraya Technological University, Belagavi.

I further declare that I have not submitted this report either in part or in full to any other university for the award of any degree.

Harsh Ticku **1BM22CS109**

Place: Bengaluru

Date: 03/03/2024

1. WRITE TO PROGRAM TO FIND QUADRATIC EQUATION OF GIVEN ROOTS.

Program:-

```
import java.util.Scanner;

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

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

   void computeRoots() {
      while (a == 0) {
         System.out.println("Not a quadratic equation");
         System.out.println("Enter a non-zero value of a:");
         Scanner scanner = new Scanner(System.in);
         a = scanner.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("Harsh Ticku");
        System.out.println("1BM22CS109");
        Quadratic quadraticEquation = new Quadratic();
        quadraticEquation.getValues();
        quadraticEquation.computeRoots();
    }
}
```

Output:-

```
C:\Users\Admin\OneDrive\Desktop\JAVA>java QuadraticMain
Harsh Ticku
1BM22CS109
Enter the values of a, b, c:
5 5 0
Roots are real and distinct
Roots are Root1=0.0 and Root2=-1.0
```

```
C:\Users\Admin\OneDrive\Desktop\JAVA>java QuadraticMain
Harsh Ticku
1BM22CS109
Enter the values of a, b, c:
1 2 3
Roots are imaginary and Root1=-1.0+i1.4142135623730951 and Root2=-1.0-i1.4142135623730951
```

```
C:\Users\Admin\OneDrive\Desktop\JAVA>java QuadraticMain
Harsh Ticku
1BM22CS109
Enter the values of a, b, c:
0 0 0
Not a quadratic equation
Enter a non-zero value of a:
10
Roots are real and equal
Roots are Root1=Root2=0.0
```

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.

Program:

```
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 BooksMain {
    public static void main(String[] args) {
        System.out.println("Harsh Ticku");
        System.out.println("1BM22CS109");

        Scanner s = new Scanner(System.in);
        int n;

        System.out.println("Enter the number of books:");
        n = s.nextInt();
        s.nextLine(); // Consume the newline left by nextInt()

        Book[] booksArray = new Book[n];

        for (int i = 0; i < n; i++) {
            System.out.println("Book " + (i + 1) + ":");
            System.out.println("Enter the book name");
            String name = s.nextLine();
```

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

            s.nextLine(); // Consume the newline left by nextInt()

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

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

Output:

```
C:\Users\Admin\OneDrive\Desktop\JAVA>java BooksMain
Harsh Ticku
1BM22CS109
Enter the number of books:
3
Book 1:
Enter the book name
Maths
Enter the author
R D Sharma
Enter the price
450
Enter the number of pages
600
Book 2:
Enter the book name
Cant hurt me
Enter the author
David Goggins'
Enter the price
300
Enter the number of pages
450
Book 3:
Enter the book name
Atomic Habits
Enter the author
James Clear
Enter the price
200
Enter the number of pages
300
Book 1
Book Name: Maths
Author Name: R D Sharma
Book Price: 450
Number of pages: 600

Book 2
Book Name: Cant hurt me
Author Name: David Goggins'
Book Price: 300
Number of pages: 450

Book 3
Book Name: Atomic Habits
Author Name: James Clear
Book Price: 200
Number of pages: 300
```

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).

Program:

```
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 + 1) + " Marks");
        for (int j = 0; j < 6; j++) {
            System.out.println("Enter Marks of Subject" + (j + 1) + ":");
            marks[j] = s.nextInt();
            percentage += marks[j];
        }
    }

    void calculatePercentage(int i) {
        percentage = (percentage / 6);

        System.out.println("Percentage of student" + (i + 1) + "=" + percentage + "%");
    }
}

class StudentMain {
    public static void main(String[] args) {
        System.out.println("Harsh Ticku");
        System.out.println("1BM22CS109");
        System.out.println("Enter the number of Students");
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        Student[] students = new Student[n];
        for (int i = 0; i < n; i++) {
            students[i] = new Student();
            students[i].getData(i);
        }
        for (int i = 0; i < n; i++) {
```

```
            students[i].calculatePercentage(i);
        }
    }
}
```

Output:

```
C:\Users\Admin\OneDrive\Desktop\JAVA>java StudentMain
Harsh Ticku
1BM22CS109
Enter the number of Students
2
Enter USN:
1BM22CS109
Enter Name:
Harsh_Ticku'
Enter Student1 Marks
Enter Marks of Subject1:
89
Enter Marks of Subject2:
78
Enter Marks of Subject3:
93
Enter Marks of Subject4:
88
Enter Marks of Subject5:
85
Enter Marks of Subject6:
87
Enter USN:
1BM22CS109
Enter Name:
Jasmeet
Enter Student2 Marks
Enter Marks of Subject1:
89
Enter Marks of Subject2:
78
Enter Marks of Subject3:
90
Enter Marks of Subject4:
99
Enter Marks of Subject5:
91
Enter Marks of Subject6:
87
Percentage of student1=86.666664%
Percentage of student2=89.0%
```

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.

Program:

```
import java.util.Scanner;

abstract class Shape {
    int dimension1;
    int dimension2;

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

    abstract void printArea();
}

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

    void printArea() {
        double area = dimension1 * dimension2;
        System.out.println("Area of rectangle = " + area);
    }
}

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

    void printArea() {
        double area = 0.5 * dimension1 * dimension2;
        System.out.println("Area of triangle = " + area);
    }
}

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

```
    void printArea() {
        double area = Math.PI * dimension1 * dimension1;
        System.out.println("Area of circle = " + area);
    }
}

class ImprovedAbstractMain {
    public static void main(String[] args) {
        System.out.println("Harsh Ticku");
        System.out.println("1BM22CS109");

        Scanner scanner = new Scanner(System.in);

        System.out.println("Enter the length and breadth of the rectangle:");
        int length = scanner.nextInt();
        int breadth = scanner.nextInt();

        System.out.println("Enter base and height of the triangle:");
        int base = scanner.nextInt();
        int height = scanner.nextInt();

        System.out.println("Enter the radius of the circle:");
        int radius = scanner.nextInt();

        Rectangle rectangle = new Rectangle(length, breadth);
        Triangle triangle = new Triangle(height, base);
        Circle circle = new Circle(radius);

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

Output:-

```
C:\Users\Admin\OneDrive\Desktop\JAVA>java ImprovedAbstractMain
Harsh Ticku
1BM22CS109
Enter the length and breadth of the rectangle:
22 67
Enter base and height of the triangle:
55 46
Enter the radius of the circle:
50
Area of rectangle = 1474.0
Area of triangle = 1265.0
Area of circle = 7853.981633974483
```

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.

Program:

```
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("Balance: $" + balance);
    }
}

class CurAcct extends Account {
    double minBalance;
    double serviceCharge;

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

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

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

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 calculateInterest() {
        double interest = balance * interestRate / 100;
        balance += interest;
        System.out.println("Interest of $" + interest + " added.");
    }

    void withdraw(double amount) {
        if (balance - amount >= 0) {
            balance -= amount;
            System.out.println("Withdrawal of $" + amount + " successful.");
        } else {
            System.out.println("Insufficient funds. Withdrawal failed.");
        }
    }
}

class Bank {
    public static void main(String[] args) {
        System.out.println("Harsh Ticku");
        System.out.println("1BM22CS109");

        Scanner scanner = new Scanner(System.in);

        // Creating savings account
        SavAcct savings = new SavAcct("John Doe", 123456, "Savings", 1000, 5); //5% interest
rate

        // Creating current account
        CurAcct current = new CurAcct("Jane Doe", 654321, "Current", 2000, 500, 10);
        // $500 minimum balance, $10 service charge

        System.out.println("Welcome to our bank!");
        while (true) {
            System.out.println("\n 1.Deposit\n 2.Withdraw\n 3.Display Balance \n 4.Exit");
            System.out.print("Enter your choice: ");
```

```
        int choice = scanner.nextInt();
        switch (choice) {
            case 1:
                System.out.print("Enter amount to deposit: ");
                double depositAmount = scanner.nextDouble();
                System.out.print("Select account (1 for Savings, 2 for Current): ");
                int accountChoice = scanner.nextInt();
                if (accountChoice == 1)
                    savings.deposit(depositAmount);
                else if (accountChoice == 2)
                    current.deposit(depositAmount);
                break;
            case 2:
                System.out.print("Enter amount to withdraw: ");
                double withdrawAmount = scanner.nextDouble();
                System.out.print("Select account (1 for Savings, 2 for Current): ");
                accountChoice = scanner.nextInt();
                if (accountChoice == 1)
                    savings.withdraw(withdrawAmount);
                else if (accountChoice == 2) {
                    current.withdraw(withdrawAmount);
                    current.deductServiceCharge();
                }
                break;
            case 3:
                System.out.print("Select account (1 for Savings, 2 for Current): ");
                accountChoice = scanner.nextInt();
                if (accountChoice == 1)
                    savings.displayBalance();
                else if (accountChoice == 2)
                    current.displayBalance();
                break;
            case 4:
                System.out.println("Thank you for banking with us!");
                System.exit(0);
            default:
                System.out.println("Invalid choice. Please try again.");
        }
    }
```

```
    }
}
```

Output:

```
C:\Users\Admin\OneDrive\Desktop\JAVA>java Bank
Harsh Ticku
1BM22CS109
Welcome to our bank!

 1.Deposit
 2.Withdraw
 3.Display Balance
 4.Exit
Enter your choice: 1
Enter amount to deposit: 25000
Select account (1 for Savings, 2 for Current): 1
Deposit of $25000.0 successful.

 1.Deposit
 2.Withdraw
 3.Display Balance
 4.Exit
Enter your choice: 1
Enter amount to deposit: 30000
Select account (1 for Savings, 2 for Current): 2
Deposit of $30000.0 successful.

 1.Deposit
 2.Withdraw
 3.Display Balance
 4.Exit
Enter your choice: 2
Enter amount to withdraw: 12000
Select account (1 for Savings, 2 forCurrent): 1
Withdrawal of $12000.0 successful.

 1.Deposit
 2.Withdraw
 3.Display Balance
 4.Exit
Enter your choice: 3
Select account (1 for Savings, 2 for Current): 1
Balance: $14000.0

 1.Deposit
 2.Withdraw
 3.Display Balance
 4.Exit
Enter your choice: 4
Thank you for banking with us!
```

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

Program:

```
// Student.java
package CIE;

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

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

// Internals.java
package CIE;

public class Internals extends Student {
    public int[] internalMarks;

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

```
// Externals.java
package SEE;

import CIE.Student;

public class Externals extends Student {
    public int[] seeMarks;

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

// Main1.java
import CIE.Internals;
import SEE.Externals;

import java.util.Scanner;

public class Main1 {
    public static void main(String[] args) {
        System.out.println("Harsh Ticku");
        System.out.println("1BM22CS109");

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the number of students: ");
        int numStudents = scanner.nextInt();

        Internals[] cieStudents = new Internals[numStudents];
        Externals[] seeStudents = new Externals[numStudents];

        // Input CIE marks
        for (int i = 0; i < numStudents; i++) {
            System.out.println("\nEnter details for CIE of student " + (i + 1));
            cieStudents[i] = getInputForCIE(scanner);
        }

        // Input SEE marks
        for (int i = 0; i < numStudents; i++) {
            System.out.println("\nEnter details for SEE of student " + (i + 1));
            seeStudents[i] = getInputForSEE(scanner);
        }

        // Displaying final marks
        System.out.println("\nFinal Marks of Students:");
        for (int i = 0; i < numStudents; i++) {
            System.out.println("\nDetails of Student " + (i + 1));
```

```
            displayStudentDetails(cieStudents[i]);
            displayStudentMarks("CIE Marks: ", cieStudents[i].internalMarks
);
            displayStudentMarks("SEE Marks: ", seeStudents[i].seeMarks);
        }
    }

    private static Internals getInputForCIE(Scanner scanner) {
        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();
        }

        return new Internals(usn, name, sem, cieMarks);
    }

    private static Externals getInputForSEE(Scanner scanner) {
        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();
        }

        return new Externals(usn, name, sem, seeMarks);
    }

    private static void displayStudentDetails(Internals student) {
        System.out.println("USN: " + student.usn);
        System.out.println("Name: " + student.name);
        System.out.println("Semester: " + student.sem);
    }

    private static void displayStudentMarks(String label, int[] marks) {
        System.out.print(label);
        for (int j = 0; j < 5; j++) {
```

```
            System.out.print(marks[j] + " ");
        }
        System.out.println();
    }
}
```

Output:

```
C:\Users\Admin\OneDrive\Desktop\JAVA\Packages>java Main1
Harsh Ticku
1BM22CS109
Enter the number of students: 2

Enter details for CIE of student 1
USN: 1BM22CS109
Name: Harsh_Ticku
Semester: 3
Enter CIE marks for 5 courses: 88 89 94 92 80

Enter details for CIE of student 2
USN: 1BM22CS132
Name: Jasmeet
Semester: 3
Enter CIE marks for 5 courses: 99 83 78 88 94

Enter details for SEE of student 1
USN: 1BM22CS109
Name: Harsh_Ticku
Semester: 3
Enter SEE marks for 5 courses: 99 84 95 87 89

Enter details for SEE of student 2
USN: 1BM22CS132
Name: Jasmeet
Semester: 3
Enter SEE marks for 5 courses: 89 99 83 90 92

Final Marks of Students:

Details of Student 1
USN: 1BM22CS109
Name: Harsh_Ticku
Semester: 3
CIE Marks: 88 89 94 92 80
SEE Marks: 99 84 95 87 89

Details of Student 2
USN: 1BM22CS132
Name: Jasmeet
Semester: 3
CIE Marks: 99 83 78 88 94
SEE Marks: 89 99 83 90 92
```

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.

Program:

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

class Parent {
    private int age;

    // Constructor for Parent class
    public Parent(int age) throws WrongAgeException {
        if (age < 0) {
            throw new WrongAgeException("Parent's age cannot be negative");
        }
        this.age = age;
    }

    // Getter method for age
    public int getAge() {
        return age;
    }
}

class Child extends Parent {
    private int childAge;

    // Constructor for Child class
    public Child(int parentAge, int childAge) throws WrongAgeException {
        super(parentAge);

        // Check if child's age is less than parent's age
        if (childAge >= parentAge) {
            throw new WrongAgeException("Child's age should be less than Parent's age");
        }

        this.childAge = childAge;
    }

    // Getter method for child's age
```

```
    public int getChildAge() {
        return childAge;
    }
}
class InheritanceExceptionDemo {
    public static void main(String[] args) {
        System.out.println("Harsh Ticku");
        System.out.println("1BM22CS109");
        try {
            // Creating an instance of Parent
            Parent parent = new Parent(40);
            System.out.println("Parent's age: " + parent.getAge());

            // Creating an instance of Child
            Child child = new Child(40, 20); // This will throw an exception if child's age is >=
parent's age
            System.out.println("Child's age: " + child.getChildAge());
        } catch (WrongAgeException e) {
            System.out.println("Exception: " + e.getMessage());
        }
    }
}
```

Output:

```
C:\Users\Admin\OneDrive\Desktop\JAVA>javac InheritanceExceptionDemo.java

C:\Users\Admin\OneDrive\Desktop\JAVA>java InheritanceExceptionDemo
Harsh Ticku
1BM22CS109
Parent's age: 40
Child's age: 20
```

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.

Program:

```
class DisplayThread extends Thread {
    private String message;
    private int intervalMillis;

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

```
    public void run() {
        while (true) {
            try {
                System.out.println(message);
                Thread.sleep(intervalMillis);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}

public class DisplayProgram {
    public static void main(String[] args) {
        DisplayThread thread1 = new DisplayThread("Harsh Ticku - BMS College of Engineering",
10000); // 10 seconds
        DisplayThread thread2 = new DisplayThread("1BM22CS109 - CSE", 2000); // 2 seconds

        thread1.setName("Thread-1");
        thread2.setName("Thread-2");

        thread1.setDaemon(true); // Set thread1 as daemon thread if needed
        thread2.setDaemon(true); // Set thread2 as daemon thread if needed

        thread1.start();
        thread2.start();

        // Optionally, add a delay to the main thread to observe the output
        try {
            Thread.sleep(30000); // Sleep for 30 seconds
        } catch (InterruptedException e) {
            e.printStackTrace();
        }

        // Optionally, stop the threads after observing the output
        thread1.interrupt();
        thread2.interrupt();
    }
}
```

Output:

```
C:\Users\Admin\OneDrive\Desktop\JAVA>java DisplayProgram
Harsh Ticku - BMS College of Engineering
1BM22CS109 - CSE
1BM22CS109 - CSE
1BM22CS109 - CSE
1BM22CS109 - CSE
1BM22CS109 - CSE
Harsh Ticku - BMS College of Engineering
1BM22CS109 - CSE
1BM22CS109 - CSE
1BM22CS109 - CSE
1BM22CS109 - CSE
1BM22CS109 - CSE
Harsh Ticku - BMS College of Engineering
1BM22CS109 - CSE
1BM22CS109 - CSE
1BM22CS109 - CSE
1BM22CS109 - CSE
1BM22CS109 - CSE
java.lang.InterruptedException: sleep interrupted
```

9. Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box.

Program:

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

public class ExceptionDemo extends JFrame implements ActionListener {
   private JTextField t1, t2, t3;
   private JLabel l1, l2;
   private JButton b1, b2;

   public ExceptionDemo() {
      setLayout(new FlowLayout());
      l1 = new JLabel("Num1 :");
      add(l1);
      t1 = new JTextField(5);
      add(t1);
      l2 = new JLabel("Num2 :");
      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); // Set an appropriate size
        setVisible(true);
    }

    public void actionPerformed(ActionEvent act) {
        String str = act.getActionCommand();
        if (str.equals("Divide")) {
            try {
                int num1 = Integer.parseInt(t1.getText());
                int num2 = Integer.parseInt(t2.getText());
                int num3 = num1 / num2;
                t3.setText("" + num3);
            } catch (ArithmeticException e) {
                JOptionPane.showMessageDialog(this, "ArithmeticException: Cannot divide by zero!");
            } catch (NumberFormatException e) {
                JOptionPane.showMessageDialog(this, "NumberFormatException: Please enter valid integers
      for Num1 and Num2.");
            }
        } else {
            t1.setText("");
            t2.setText("");
            t3.setText("");
        }
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                new ExceptionDemo();
            }
});}}
```

Output: