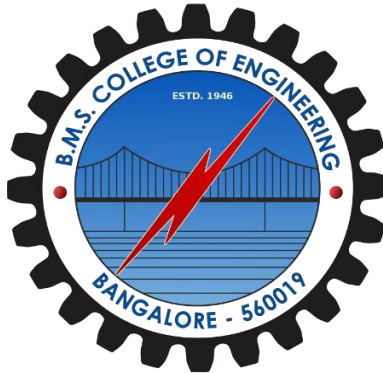# B.M.S. College of Engineering

(Autonomous College Affiliated to Visvesvaraya Technological University, Belgaum)

Bull Temple Road, Basavanagudi, Bengaluru – 560019

Department of

Computer Science & Engineering (CSE)

Lab Programs Report
Course Title: Object Oriented Java Programming
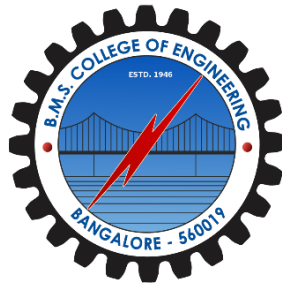
Course Code: 23CS3PCOOJ

BY

Ruth Mary Paul (1BM22CS360)
B.M.S. College of Engineering

(Autonomous College Affiliated to Visvesvaraya Technological University, Belgaum)

Bull Temple Road, Basavanagudi, Bengaluru – 560019

Department of
Computer Science & Engineering (CSE)

CERTIFICATE

This is to certify that the report on "**Java Lab Programs**" has been carried out by **Ruth Mary Paul** bearing USN **1BM22CS360** as a part of AAT for the course **Object Oriented Java Programming** with course code **23CS3PCOOJ**,

Computer Science and Engineering from Visvesvaraya Technological University,

Belgaum during the year 2023–24. It is certified that all corrections/suggestions

indicated for Internal Assessments have been incorporated in the report.

Ruth Mary Paul   **Shravya AR**
1BM22CS360   Assistant Professor
                                                               Department of CSE
                                                               BMSCE, Bengaluru-19

Table of contents

LAB PROGRAM 1

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 percentage of a student.

CODE

```java
import java.util.Scanner;

class Student {
   private String usn;
private String name;
private int[] marks;

   // Constructor    public Student(String usn,
String name) {       this.usn = usn;        this.name
= name;        this.marks = new int[6]; // Assuming 6
subjects
   }

   // Method to accept student details
public void acceptDetails() {
      Scanner scanner = new Scanner(System.in);
System.out.print("Enter USN: ");        this.usn =
scanner.nextLine();       System.out.print("Enter
Name: ");
      this.name = scanner.nextLine();

      for (int i = 0; i < marks.length; i++) {
         System.out.print("Enter marks for Subject " + (i + 1) + ": ");
this.marks[i] = scanner.nextInt();
      }
   }

   // Method to calculate percentage
public double calculatePercentage() {
int totalMarks = 0;        for (int mark :
marks) {          totalMarks += mark;
      }

      return (double) totalMarks / marks.length;
   }
```

```java
    // Method to display student details
public void displayDetails() {
      System.out.println("USN: " + this.usn);
System.out.println("Name: " + this.name);

      // Displaying marks for each subject
System.out.print("Marks: ");        for (int i
= 0; i < marks.length; i++) {
        System.out.print("Subject " + (i + 1) + ": " + marks[i] + "  ");
    }
    System.out.println();
    System.out.println("Percentage: " + calculatePercentage() + "%");
  }
}




public class StudentRun {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
      // Number of students
      System.out.print("Enter the number of students: ");
int numStudents = scanner.nextInt();       // Creating
an array of Student objects      Student[] students =
new Student[numStudents];

      // Accepting details for each student
for (int i = 0; i < numStudents; i++) {
        System.out.println("\nEnter details for Student " + (i + 1) + ":");
students[i] = new Student("", "");          students[i].acceptDetails();
    }

       // Displaying details for each student
System.out.println("\nDetails of Students:");        for
(Student student : students) {
student.displayDetails();
        System.out.println("\n-----------------------");
    }
  }
}
```

OUTPUT

Lab program 2

Develop a Java program that prints all real solutions to the quadratic equation ax2+bx+c = 0. Read in a, b, c and use the quadratic formula. If the discriminate b2-4ac is negative, display a message stating that there are no real solutions.

CODE

```
import java.util.Scanner;

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

      // Input coefficients a, b, and c
System.out.print("Enter coefficient a: ");       double
```
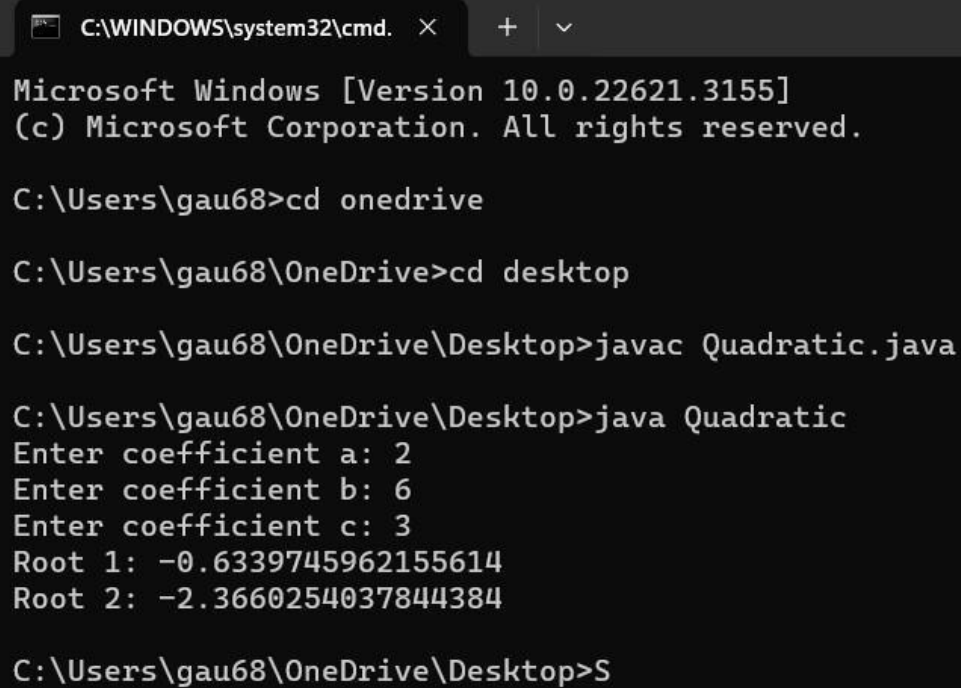
5

```java
a = scanner.nextDouble();
System.out.print("Enter coefficient b: ");        double
b = scanner.nextDouble();
System.out.print("Enter coefficient c: ");
      double c = scanner.nextDouble();

      // Calculate the discriminant
double discriminant = b * b - 4 * a * c;

      // Check the discriminant for roots        if (discriminant
> 0) {           // Two distinct real roots          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) {
// One real root (double root)
double root = -b / (2 * a);
          System.out.println("Root: " + root);
      } else {
          // Complex roots
double realPart = -b / (2 * a);
          double imaginaryPart = Math.sqrt(-discriminant) / (2 * a);
          System.out.println("Root 1: " + realPart + " + " + imaginaryPart + "i");
          System.out.println("Root 2: " + realPart + " - " + imaginaryPart + "i");
      }
   }
}
```

OUTPUT


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

C:\Users\gau68>cd onedrive

C:\Users\gau68\OneDrive>cd desktop

C:\Users\gau68\OneDrive\Desktop>javac Quadratic.java

C:\Users\gau68\OneDrive\Desktop>java Quadratic
Enter coefficient a: 2
Enter coefficient b: 6
Enter coefficient c: 3
Root 1: -0.6339745962155614
Root 2: -2.3660254037844384

C:\Users\gau68\OneDrive\Desktop>S
```

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;

class Books {
    String   Name;
    String     Author;
    int  price;        int
    numPages;

    Books(String Name, String Author, int price, int numPages) {
        this.Name = Name;        this.Author = Author;
        this.numPages = numPages;        this.price = price;
    }

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

class BooksRun {    public static void
main(String[] args) {        Scanner s = new
Scanner(System.in);
        int n;
```

```java
        String   Name;
String        Author;
int  price;              int
numPages;
        System.out.print("Enter the number of books: ");
n = s.nextInt();
        Books b[] = new Books[n];
         for (int i = 0; i < n; i++)
{
            System.out.println(" ");
            System.out.print("Enter name of book: ");
            Name = s.next();
            System.out.print("Enter name of Author: ");
            Author = s.next();
            System.out.print("Enter price: ");
price = s.nextInt();
            System.out.print("Enter numPages: ");
numPages = s.nextInt();

            b[i] = new Books(Name, Author, price, numPages);
        }        for (int i = 0; i <
n; i++) {
            System.out.println("\nBook" + (i + 1));
            System.out.println(b[i].toString());
        }
    }
}
```

OUTPUT

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

```java
import java.util.Scanner;

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

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

    public abstract void printArea();
}

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

    public void printArea() {
       int area = dimension1 * dimension2;
       System.out.println("Area of Rectangle: " + area);
    }}

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

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

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

```java
import java.util.Scanner;
abstract class Shape {
 protected int dimension1;
 protected int dimension2;
 public Shape(int dimension1, int dimension2) {
 this.dimension1 = dimension1;
 this.dimension2 = dimension2;
 }
 public abstract void printArea();
}
class Rectangle extends Shape {
 public Rectangle(int length, int width) {
 super(length, width);
 }
 public void printArea() {
 int area = dimension1 * dimension2;
 System.out.println("Area of Rectangle: " + area);
 }
}
class Triangle extends Shape {
 public Triangle(int base, int height) {
 super(base, height);
 }
 public void printArea() {
 double area = 0.5 * dimension1 * dimension2;
 System.out.println("Area of Triangle: " + area);
 }
}
class Circle extends Shape {
 public Circle(int radius) {
 super(radius, 0); // Only one dimension (radius) needed for a circle
```

```java
  }
  public void printArea() {
  double area = Math.PI * dimension1 * dimension1;
  System.out.println("Area of Circle: " + area);
  }
}
public class Main {
 public static void main(String[] args) {
 Rectangle rectangle = new Rectangle(4, 5);
 rectangle.printArea();
 Triangle triangle = new Triangle(3, 6);
 triangle.printArea();
 Circle circle = new Circle(7);
 circle.printArea();
 }
}
```

OUTPUT


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

C:\Users\gau68>cd onedrive

C:\Users\gau68\OneDrive>cd desktop

C:\Users\gau68\OneDrive\Desktop>javac Main.java

C:\Users\gau68\OneDrive\Desktop>java Main
Area of Rectangle: 20
Area of Triangle: 9.0
Area of Circle: 153.93804002589985

C:\Users\gau68\OneDrive\Desktop>
```

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

```java
import java.util.Scanner;

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

        System.out.println("Welcome to the Bank!");

        System.out.print("Enter name for Account (Savings/Current): ");
        String savingsAccountName = scanner.nextLine();
    String currentAccountName = savingsAccountName;

    // Get user input for Savings Account
        System.out.print("Enter account number for Savings Account: ");
        String savingsAccountNumber = scanner.nextLine();
        SavingsAccount savingsAccount = new SavingsAccount(savingsAccountName,
savingsAccountNumber);

        // Get user input for Current Account
        System.out.print("Enter account number for Current Account: ");
        String currentAccountNumber = scanner.nextLine();
        CurrentAccount currentAccount = new CurrentAccount(currentAccountName,
currentAccountNumber);

        // Perform operations on savings account
        System.out.println("\nPerforming operations on Savings Account:");
System.out.print("Enter amount to deposit into Savings Account: ");        double
savingsDepositAmount = scanner.nextDouble();
savingsAccount.deposit(savingsDepositAmount);
savingsAccount.displayBalance();

        System.out.print("Enter amount to withdraw from Savings Account: ");
double savingsWithdrawAmount = scanner.nextDouble();
savingsAccount.withdraw(savingsWithdrawAmount);
savingsAccount.displayBalance();

        // Perform operations on current account
        System.out.println("\nPerforming operations on Current Account:");
System.out.print("Enter amount to deposit into Current Account: ");        double
currentDepositAmount = scanner.nextDouble();
currentAccount.deposit(currentDepositAmount);
currentAccount.displayBalance();
```

```java
        System.out.print("Enter amount to withdraw from Current Account: ");
double currentWithdrawAmount = scanner.nextDouble();
currentAccount.withdraw(currentWithdrawAmount);
currentAccount.displayBalance();

        scanner.close();
    } }

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

    public Account(String customerName, String accountNumber) {
this.customerName = customerName;        this.accountNumber =
accountNumber;
        this.balance = 0;
    }

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

    public void displayBalance() {
        System.out.println("Account Number: " + accountNumber + "\nBalance: $" + balance);
    } }

class SavingsAccount extends Account {
    public SavingsAccount(String customerName, String accountNumber) {
super(customerName, accountNumber);
    }

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

class CurrentAccount extends Account {     private double minimumBalance = 1000; //
Assuming a minimum balance requirement
```

```java
    public CurrentAccount(String customerName, String accountNumber) {
super(customerName, accountNumber);
    }

    public void withdraw(double amount) {        if
(balance - amount >= minimumBalance) {
        balance -= amount;
        System.out.println("Withdrawal of $" + amount + " successful.");
    } else {
        System.out.println("Insufficient funds. Service charge applied.");
imposeServiceCharge();
    }
}

    private void imposeServiceCharge() {        double serviceCharge
= 20; // Assuming a service charge of $20        balance -=
serviceCharge;
    System.out.println("Service charge of $" + serviceCharge + " imposed.");
    }
}
```

OUTPUT


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

C:\Users\gau68>cd onedrive

C:\Users\gau68\OneDrive>cd desktop

C:\Users\gau68\OneDrive\Desktop>javac Bank.java

C:\Users\gau68\OneDrive\Desktop>java Bank
Welcome to the Bank!
Enter name for Account (Savings/Current): Savings
Enter account number for Savings Account: 123
Enter account number for Current Account: 345

Performing operations on Savings Account:
Enter amount to deposit into Savings Account: 5000
Deposit of $5000.0 successful.
Account Number: 123
Balance: $5000.0
Enter amount to withdraw from Savings Account: 567
Withdrawal of $567.0 successful.
Account Number: 123
Balance: $4433.0

Performing operations on Current Account:
Enter amount to deposit into Current Account: 5600
Deposit of $5600.0 successful.
Account Number: 345
Balance: $5600.0
Enter amount to withdraw from Current Account: 4500
Withdrawal of $4500.0 successful.
Account Number: 345
Balance: $1100.0

C:\Users\gau68\OneDrive\Desktop>
```

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

```
package CIE;
public class student
{
public String usn,name; public
int sem;
public student(String usn,String name,int sem)
{this.usn=usn; this.name=name;
this.sem=sem;
}
}

package CIE; public class Internals
extends student
{
public int m[]=new int[5]; public Internals(String
usn,String name,int sem,int[] m)
{super(usn,name,sem);
this.m=m;
}
}

package SEE; import
CIE.student;
public class Externals extends student
{
public int sm[]=new int[5]; public Externals(String
usn,String name,int sem,int[] sm)
{super(usn,name,sem);
this.sm=sm;
}
```

```java
}

import java.util.*; import
CIE.student; import
CIE.Internals; import
SEE.*;
public class Main
{
public static void main(String args[])
{
int fm=0;
Scanner in=new Scanner(System.in); System.out.print("Enter
no. of students: ");
int n=in.nextInt(); Internals
[]im=new Internals[n];
Externals []em=new
Externals[n]; student
[]stu=new student[n];
for(int i=0;i<n;i++)
{
System.out.print("\nEnter details of student "+(i+1)+": "); System.out.print("\nEnter
name: ");
in.nextLine(); String
name=in.nextLine();
System.out.print("Enter usn: ");
String usn=in.nextLine();
System.out.print("Enter semester: ");
int sem=in.nextInt(); int
[]imarks=new int[5]; int
[]emarks=new int[5];
System.out.print("Enter marks details: ");
for(int j=0;j<5;j++)
{
System.out.print("Enter internal marks for course"+(j+1)+": "); imarks[j]=in.nextInt();
System.out.print("Enter see marks for course"+(j+1)+": "); emarks[j]=in.nextInt();
}
stu[i]=new student(usn,name,sem); im[i]=new
Internals(usn,name,sem,imarks);
em[i]=new Externals(usn,name,sem,emarks);
}
System.out.print("Final marks details: ");
for(int i=0;i<n;i++)
{
System.out.println("Student"+(i+1)+": ");
```

```java
System.out.println("Name: "+stu[i].name);
System.out.println("USN: "+stu[i].usn); System.out.println("Semester:
"+stu[i].sem);
for(int j=0;j<5;j++)
{
fm+=im[i].m[j]+em[i].sm[j];
System.out.println("Final marks of course"+(j+1)+":"+fm); fm=0;
}
System.out.println();
}
}
}
```

OUTPUT


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

C:\Users\gau68>cd onedrive

C:\Users\gau68\OneDrive>cd desktop

C:\Users\gau68\OneDrive\Desktop>javac Main.java

C:\Users\gau68\OneDrive\Desktop>java Main
Enter no. of students: 1

Enter details of student 1:
Enter name: gaurav
Enter usn: 45678
Enter semester: 3
Enter marks details: Enter internal marks for course1: 34
Enter see marks for course1: 98
Enter internal marks for course2: 39
Enter see marks for course2: 97
Enter internal marks for course3: 33
Enter see marks for course3: 56
Enter internal marks for course4: 33
Enter see marks for course4: 64
Enter internal marks for course5: 40
Enter see marks for course5: 90
Final marks details: Student1:
Name: gaurav
USN: 45678
Semester: 3
Final marks of course1:132
Final marks of course2:136
Final marks of course3:89
Final marks of course4:97
Final marks of course5:130
```

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.
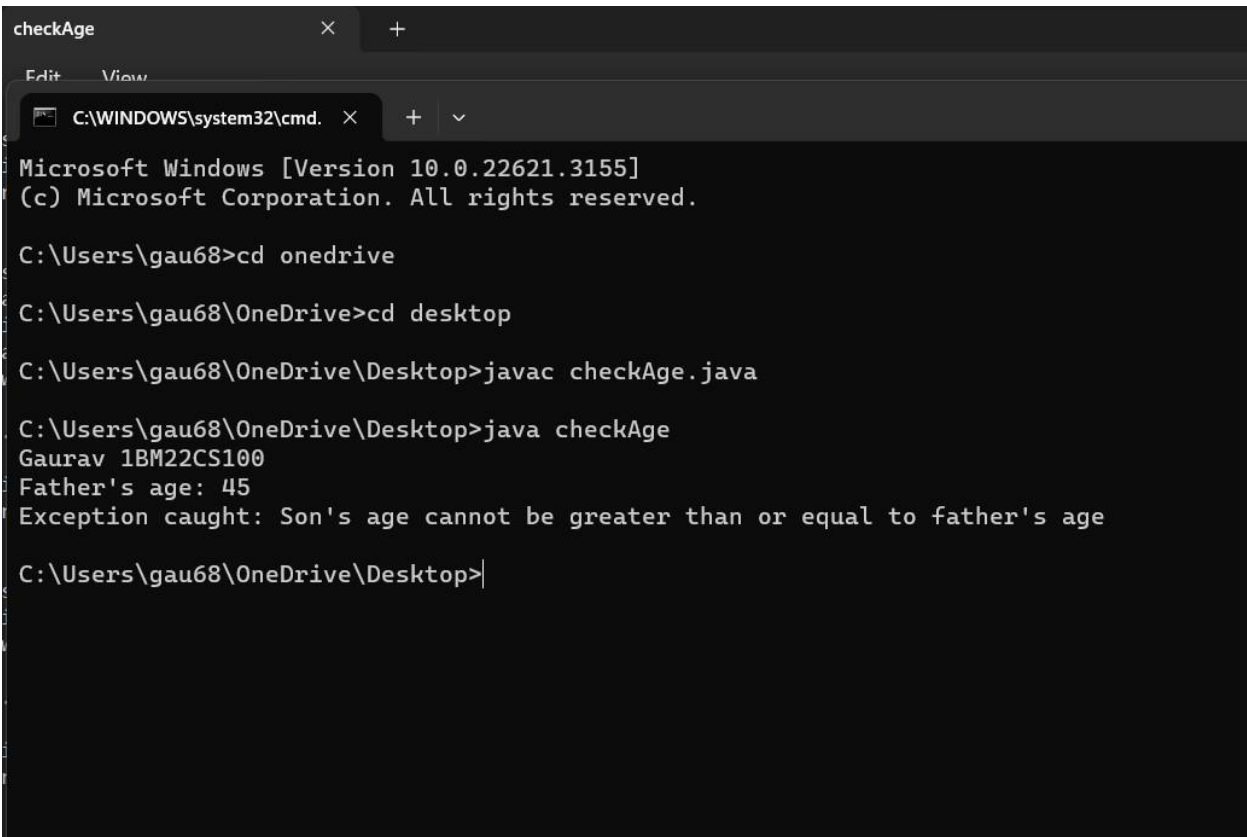
CODE

```java
class WrongAge extends Exception { public
WrongAge(String message) {
super(message);
}
}
class Father { int
age;
public Father(int age) throws WrongAge { if
(age < 0) {
throw new WrongAge("Age cannot be negative");
}
this.age = age;
}
public int getAge() {
return age;
}
}
class Son extends Father { int sonAge;
public Son(int fatherAge, int sonAge) throws WrongAge { super(fatherAge); if (sonAge >=
fatherAge) {
throw new WrongAge("Son's age cannot be greater than or equal to father's age");
}
this.sonAge = sonAge;
}
public int getSonAge() {
return sonAge;
}
}
public class checkAge { public static void main(String[] args) { System.out.println("Gaurav
1BM22CS100"); try {
Father father = new Father(45);
System.out.println("Father's age: " + father.getAge());
Son son = new Son(45, 65);
System.out.println("Son's age: " + son.getSonAge());
} catch (WrongAge e) {
System.out.println("Exception caught: " + e.getMessage());
```

```
}}
}
```
OUTPUT

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 display implements Runnable
{
String message;
int interval;
public display(String message, int interval) {this.message=message;
this.interval=interval;
}
@Override
public void run()
{try
{while(true)
{System.out.println(message);
Thread.sleep(interval);
}
}
catch(InterruptedException e)
{
System.out.println(e);
} }
}

class displayrun
{
public static void main(String args[])
{
Thread t1=new Thread (new display("BMSCE",10000));
t1.start();
Thread t2=new Thread(new display("CSE",2000));
t2.start();
}
}
```

OUTPUT