

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 CHRIS DINESH THOMAS 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: CHRIS DINESH THOMAS

USN No.: 1BM22CS080 Semester: III Section: B

Marks	
Max. Marks	Obtained
10	
Marks in Words	

Signature of the staff in-charge
Date:

Head of the Department

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

Ans:

```
import java.util.Scanner;

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

    void getd() {
        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);
        }
    }
}
```

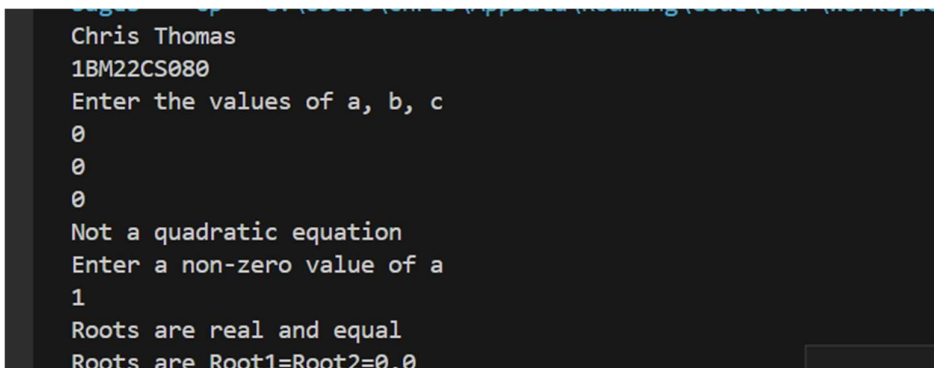
```

    } 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("Chris Thomas");
        System.out.println("1BM22CS080");
        quadratic q = new quadratic();
        q.getd();
        q.compute();
    }
}

```

Output:



```

Chris Thomas
1BM22CS080
Enter the values of a, b, c
0
0
0
Not a quadratic equation
Enter a non-zero value of a
1
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.

Ans:

```
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.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("Chris Thomas");
System.out.println("1BM22CS080");
Scanner s = new Scanner(System.in);
int n;
String name;
String author;
int price;
int numPages;

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

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

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

```

}

Output:

```
Chris Thomas
1BM22CS080
Enter the number of books:
2
Book 1:
Enter the book name
abc
Enter the author
a
Enter the price
234
Enter the number of pages
2345
Book 2:
Enter the book name
qwerty
Enter the author
rewq
Enter the price
555
Enter the number of pages
234
Book 1
Book Name: abc
Author Name: a
Book Price: 234
Number of pages: 2345
```

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).

Ans:

```
import java.util.Scanner;
```

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

    void getd(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 + "%");
    }
}

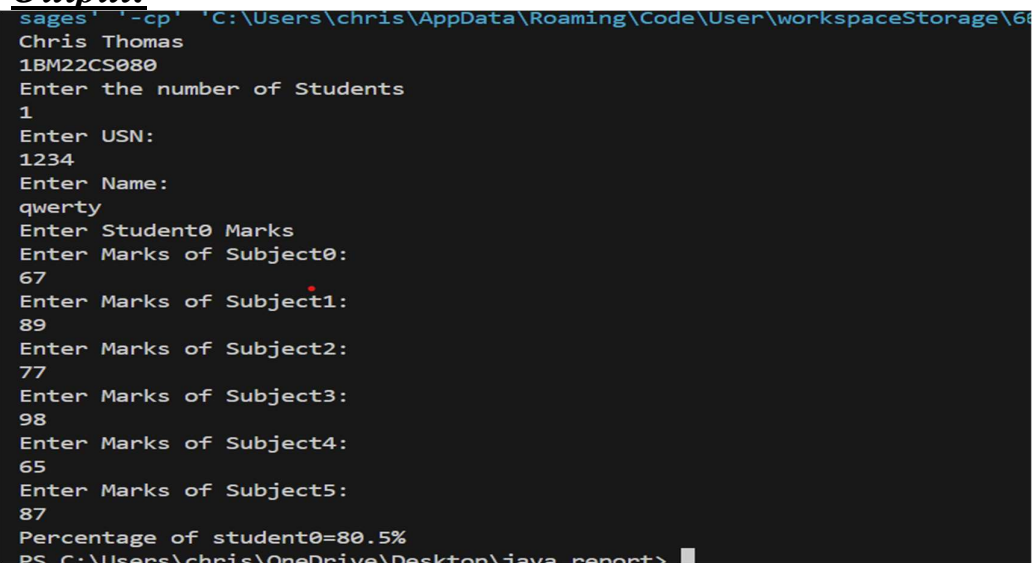
```

```

class studentMain {
    public static void main(String[] args) {
        System.out.println("Chris Thomas");
        System.out.println("1BM22CS080");
        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].getd(i);
        }
        for (int i = 0; i < n; i++) {
            s[i].calculatePercentage(i);
        }
    }
}

```

Output:



```

sages> -cp 'C:\Users\chris\AppData\Roaming\Code\User\workspaceStorage\6
Chris Thomas
1BM22CS080
Enter the number of Students
1
Enter USN:
1234
Enter Name:
qwerty
Enter Student0 Marks
Enter Marks of Subject0:
67
Enter Marks of Subject1:
89
Enter Marks of Subject2:
77
Enter Marks of Subject3:
98
Enter Marks of Subject4:
65
Enter Marks of Subject5:
87
Percentage of student0=80.5%
PS C:\Users\chris\OneDrive\Desktop> java report>

```

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.

Ans:

```
import java.util.Scanner;

abstract class shape {
    int dim1;
    int dim2;

    shape(int dim1, int dim2) {
        this.dim1 = dim1;
        this.dim2 = dim2;
    }

    abstract void printArea();
}

class rectangle extends shape {
    rectangle(int length, int breadth) {
        super(length, breadth);
    }

    void printArea() {
        double area = dim1 * dim2;
        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 * dim1 * dim2;
        System.out.println("Area of triangle = " + area);
    }
}

```

```

class circle extends shape {
    circle(int radius) {
        super(radius, 0);
    }

    void printArea() {
        double area = Math.PI * dim1 * dim1;
        System.out.println("Area of circle = " + area);
    }
}

```

```

public class AbstractMain {
    public static void main(String[] args) {
        System.out.println("Chris Thomas");
        System.out.println("1BM22CS080");
        Scanner s = new Scanner(System.in);

        System.out.println("Enter the length and breadth of the
rectangle");
        int l = s.nextInt();
        int b = s.nextInt();
    }
}

```

```

        System.out.println("Enter base and height of the triangle");
        int ba = s.nextInt();
        int h = s.nextInt();

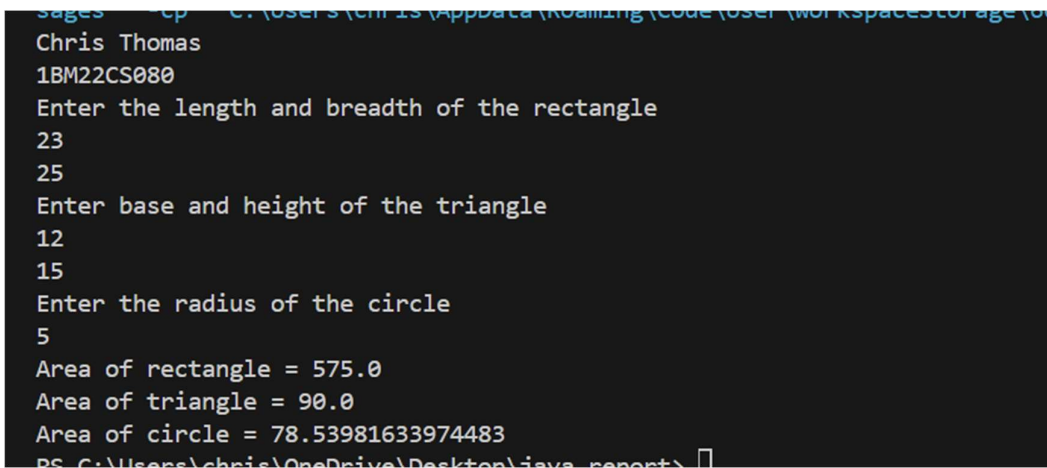
        System.out.println("Enter the radius of the circle");
        int r = s.nextInt();

        rectangle re = new rectangle(l, b);
        triangle t = new triangle(h, ba);
        circle c = new circle(r);

        re.printArea();
        t.printArea();
        c.printArea();
    }
}

```

Output:



The screenshot shows a terminal window with the following output:

```

sages - cp C:\Users\Chris\AppData\Roaming\Code\User\workspaceStorage\0...
Chris Thomas
1BM22CS080
Enter the length and breadth of the rectangle
23
25
Enter base and height of the triangle
12
15
Enter the radius of the circle
5
Area of rectangle = 575.0
Area of triangle = 90.0
Area of circle = 78.53981633974483
PS C:\Users\chris\OneDrive\Desktop\java_reports>

```

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.**

Ans:

```
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) {
        if (amount > 0) {
            balance += amount;
            System.out.println("Deposit of INR " + amount + "
successful");
        } else {
            System.out.println("Invalid deposit amount. Please enter a
positive value.");
        }
    }

    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;
    }
}
}

```

```

public class Bank {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        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: ");
    while (!scanner.hasNextInt()) {
        System.out.println("Invalid input. Please enter a
number.");
        scanner.next();
    }
    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:

```

sages' '-cp' 'C:\Users\chris\AppData\Roaming\Code\User\workspaceSto
Enter the number of users: 1

User 1
Enter customer name: qwerty
Enter account number: 12345
Enter initial deposit amount: INR 5000
Enter account type (Savings/Current): Savings

Choose an option:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings only)
5. Exit
Enter your choice: 2
Enter account number: 12345
Enter withdrawal amount: INR 3000
Withdrawal of INR 3000.0 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: 12345
Account Number: 12345
Customer Name: qwerty
Account Type: Savings
Balance: INR 2000.0

```

```

Choose an option:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings only)
5. Exit
Enter your choice: 1
Enter account number: 12345
Enter deposit amount: INR 4000
Deposit of INR 4000.0 successful

Choose an option:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings only)
5. Exit
Enter your choice: 5

```

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**

Ans:

//CIE PACKAGE

//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;
    }
}

//SEE Package

//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("Chris Thomas");
        System.out.println("1BM22CS080");

        Scanner scanner = new Scanner(System.in);

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

        internals[] cieStudents = new internals[n];
        externals[] seeStudents = new externals[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);
    }

    // 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 externals(usn, name, sem,
seeMarks);
    }

    // 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:

```

Chris Thomas
1BM22CS080
Enter the number of students: 1
Enter details for CIE of student 1
USN: 1222
Name: addff
Semester: 3
Enter CIE marks for 5 courses: 23
23
45
45
56
Enter details for SEE of student 1
USN: 1err
Name: dwwfg
Semester: 3
Enter SEE marks for 5 courses: 96
96

98
00
99

Final Marks of Students:

Details of Student 1
USN: 1222
Name: addff
Semester: 3
CIE Marks:
23 23 45 45 56
SEE Marks:
96 96 98 0 99

```


Dept. of CSE, B.M.S College of Engineering

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.

Ans:

```
class WrongAgeException extends Exception {
    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;
    }
}

```

```

public class InheritanceExceptionDemo {
    public static void main(String[] args) {
        System.out.println("Chris Thomas");
        System.out.println("1BM22CS080");
        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 due to son's age being >= father's age
            System.out.println("Son's age: " + son.getSonAge());
        } catch (WrongAgeException e) {
            System.out.println("Exception: " + e.getMessage());
        }
    }
}

```

```
}
```

Output:

```
java report_3334f46b\bin' 'InheritanceExceptionDemo'  
Chris Thomas  
1BM22CS080  
Father's age: 40  
Son'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.

Ans:

```
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) {
        System.out.println("Chris Thomas");
        System.out.println("1BM22CS080");
        DisplayThread thread1 = new DisplayThread("BMS College
of Engineering", 10000); // 10 seconds
        DisplayThread thread2 = new DisplayThread("CSE",
2000); // 2 seconds

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

```

Output:

```

Chris Thomas
1BM22CS080
BMS College of Engineering
CSE
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

```

