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 _____Suhas Kumar Reddy Nare_____ has satisfactorily completed the course of Experiments in Practical <u>OBJECT ORIENTED JAVA PROGRAMMING</u> prescribed by the Department during the odd semester 2023-24.

Name of the Candidate: _____Suhas Kumar Reddy Nare_____

USN No.: **1BM22CS294**   Semester: **III**   Section: **F**

Marks	
Max. Marks	Obtained
10	
Marks in Words	

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

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

Ans:

```java
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);
```

```java
        } 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("Suhas Kumar
        Reddy Nare");
        System.out.println("1BM22CS294");
        quadratic q = new quadratic();
        q.getd();
    } q.compute();
}
```

Output:


```
Suhas Kumar Reddy Nare
1BM22CS294
Enter the values of a, b, c

2
2
2
Roots are imaginary and Root1=-0.5+i0.8660254037844386 and Root2=-0.5-i0.8660254037844386
```

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.

<u>*Ans:*</u>

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) {

```java
System.out.println("Suhas Kumar Reddy
Nare");
System.out.println("1BM22CS294");
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:

```
Suhas Kumar Reddy Nare
1BM22CS294
Enter the number of books:
2
Book 1:
Enter the book name
harry pottter
Enter the author
jk rowling
Enter the price
520
Enter the number of pages
500
Book 2:
Enter the book name
percy jackson
Enter the author
rick riordan
Enter the price
580
Enter the number of pages
520
Book 1
Book Name: harry pottter
Author Name: jk rowling
Book Price: 520
Number of pages: 500

Book 2
Book Name: percy jackson
Author Name: rick riordan
Book Price: 580
Number of pages: 520
```

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).

<u>Ans:</u>

```java
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);
```

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

class studentMain {
    public static void main(String[] args) {
        System.out.println("Suhas Kumar");
        System.out.println("1BM22CS294");
        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:

```
Suhas Kumar Reddy Nare
1BM22CS294
Enter the number of Students
1
Enter USN:
294
Enter Name:
suhas
Enter Student0 Marks
Enter Marks of Subject0:
86
Enter Marks of Subject1:

79
Enter Marks of Subject2:
98
Enter Marks of Subject3:
94
Enter Marks of Subject4:
74
Enter Marks of Subject5:
64
Percentage of student0=82.5%
```

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.

<u>*Ans:*</u>

```java
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);
    }
}
```

```java
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("Suhas                    Kumar");
        System.out.println("1BM22CS294");
        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();
```

```java
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:

```
Suhas Kumar Reddy Nare
1BM22CS294
Enter the length and breadth of the rectangle
20
30
Enter base and height of the triangle
23
24
Enter the radius of the circle
24
Area of rectangle = 600.0
Area of triangle = 276.0
Area of circle = 1809.5573684677208
```

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;
```

```java
        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() {
    }
```

```java
    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) {
```

```java
        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")) {
```

```java
            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) {
```

```java
                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();
            }
        }
```

```java
                    break;
                case 5:
                    exit = true;
                    break;
                default:
                    System.out.println("Invalid choice. Please enter a
valid option.");
                }
            }


        }
}
```

Output:

```
Enter the number of users: 2

User 1
Enter customer name: suhas
Enter account number: 123
Enter initial deposit amount: INR 7000
Enter account type (Savings/Current):Savings

User 2
Enter customer name: user
1
Enter account number: 724
Enter initial deposit amount: INR 8600
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: 724
Enter deposit amount: INR 500
Deposit of INR 500.0successful

Choose an option:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings only)
5. Exit
Enter your choice: 2
Enter account number: 724
Enter withdrawal amount: INR 4560
Withdrawal of INR 4560.0successful

Choose an option:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings only)
5. Exit
Enter your choice: 3
Enter account number: 123
```

```
Choose an option:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings only)
5. Exit
Enter your choice: 3
Enter account number: 123
Account Number: 123
Customer Name: suhas
Account Type: Savings
Balance: INR 7000.0

Choose an option:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings only)
5. Exit
Enter your choice: 3
Enter account number: 724
Account Number: 724
Customer Name: 1
Account Type: Current
Balance: INR 4540.0

Choose an option:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings only)
5. Exit
Enter your choice:  4
Enter account number (for Savings account): 724

Choose an option:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings only)
5. Exit
Enter your choice: 4
Enter account number (for Savings account): 123
Interest of INR 350.0 added to the account
```

```
Choose an option:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings only)
5. Exit
Enter your choice: 4
Enter account number (for Savings account): 123
Interest of INR 350.0 added to the account

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;

```java
package CIE;
public class Student{
    private String usn;
    private String name;
    private int sem;
    public Student(String usn,String name,int sem){
        this.sem=sem;
        this.name=name;
        this.usn=usn;
    }
public String getUsn() {
return usn;
}
public void setUsn(String usn) {
this.usn = usn;
}
public String getName() {
return name;
}
public void setName(String name) {
this.name = name;
}
public int getSem() {
return sem;
}
public void setSem(int sem) {
this.sem = sem;
}
@Override
public String toString() {
return "Student [usn=" + usn + ", name=" + name + ", sem=" + sem + "]";
}

}

package CIE;

import java.util.Arrays;

public class Internals extends Student{
private Object[][] cieMarks;
    public Internals(String usn,String name,int sem,Object[][] cieMarks){
        super(usn,name,sem);
        this.cieMarks=cieMarks;
    }
public Object[][] getCieMarks() {
return cieMarks;
}
public void setCieMarks(Object[][] cieMarks) {
this.cieMarks = cieMarks;
}
```

```java
package SEE;

import java.util.Arrays;

import CIE.Student;

public class External extends Student{
private Object[][] seeMarks;
    public External(String usn,String name,int sem,Object[][] seeMarks){
        super(usn,name,sem);
        this.seeMarks=seeMarks;
        }
public Object[][] getSeeMarks() {
return seeMarks;
}
public void setSeeMarks(Object[][] seeMarks) {
this.seeMarks = seeMarks;
}
}
```

```java
package RESULTS;

import java.util.Arrays;
import CIE.Internals;
import CIE.Student;
import SEE.External;

public class Result {
private Internals internals;
private External external;
private Student student;
private Object[] finalMarks;
public Result(Internals internals,External external) {
this.internals=internals;
this.external=external;
this.student=new Student(internals.getName(),internals.getUsn(),internals.getSem());
}
public Object[]getFinalMarks() {
finalMarks=new Object[internals.getCieMarks().length];
int i=0;
for(Object[] cieEntry :internals.getCieMarks()) {
int marks=0;
for(Object[] seeEntry :external.getSeeMarks()) {
if(cieEntry[0]==seeEntry[0]) {
marks=(int)cieEntry[1]+(int)seeEntry[1];

}
}
Object[] totalMarks= {cieEntry[0],marks};
finalMarks[i]= totalMarks;
i++;

    }

return finalMarks;
}
@Override
public String toString() {
return "Result [\n" + internals + "\n" + external + "\nstudent=" + student + "\nfinalMarks="
+Arrays.deepToString(finalMarks) + "\n]";
}




}
```

Output:

```
Result [
Internals [cieMarks=[[ds, 45], [m1, 35], [oop, 47], [java, 32], [coa, 43]]]
External [seeMarks=[[ds, 35], [m1, 45], [oop, 37], [java, 32], [coa, 43]]]
student=Student [usn=Suhas KRN, name=22cs294, sem=3]
finalMarks=[[ds, 80], [m1, 80], [oop, 84], [java, 64], [coa, 86]]
]
```

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;
    }
}
```

```java
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) {
        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:

```
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:

```java
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("BMS College
of Engineering", 10000); // 10 seconds
        DisplayThread thread2 = new DisplayThread("CSE",
2000); // 2 seconds

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

Output:

```
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
```