19CSE204 – Object Oriented Paradigm

LAB REPORT – 1

BASIC JAVA PROGRAMS

SUBMITTED BY

NEESHNA LAKSHMI H – CH.EN.U4CCE22049

SUBMITTED TO

Dr. S SUTHIR

Program 1: Scanner Input/Output

Objective:

To take input from the user and display student information using the Scanner class.

Code:

```
import java.util.Scanner;

public class StudentDetails {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter Name: ");

        String name = sc.nextLine();

        System.out.print("Enter Roll No: ");

        String rollNo = sc.nextLine();

        System.out.print("Enter Department: ");

        String dept = sc.nextLine();

        System.out.print("Enter Course: ");

        String course = sc.nextLine();

        System.out.print("Enter Subject: ");

        String subject = sc.nextLine();


        System.out.println("\n--- Student Details ---");

        System.out.println("Name: " + name);

        System.out.println("Roll No: " + rollNo);

        System.out.println("Department: " + dept);

        System.out.println("Course: " + course);

        System.out.println("Subject: " + subject);

    }

}
```

Output:

```
D:\JAVA\JAVA_PROGRAMS>java StudentDetails
Enter your Name         : NEESHNA LAKSHMI H
Enter your Age          : 20
Enter your Roll Number  : CH.EN.U4CCE22049
Enter your Department   : CCE
Enter your Course       : 19CSE204
Enter your Subject      : OOPS

========== STUDENT DETAILS ==========
Name         : NEESHNA LAKSHMI H
Age          : 20
Roll Number  : CH.EN.U4CCE22049
Department   : CCE
Course       : 19CSE204
Subject      : OOPS
=====================================
```

Program 2: All Operators

Objective:

To perform various arithmetic, relational, logical, and unary operations on subject marks entered by the user.

Code:

```
import java.util.Scanner;


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


        System.out.print("Enter marks in Subject 1: ");
        int sub1 = sc.nextInt();


        System.out.print("Enter marks in Subject 2: ");
        int sub2 = sc.nextInt();


        // Arithmetic Operators
        int total = sub1 + sub2;      // +
        int difference = sub1 - sub2; // -
```

```
int product = sub1 * sub2;    // *
int average = total / 2;      // /
int mod = total % 100;        // %

// Unary Operators
int studentCount = 0;
++studentCount;               // ++
--studentCount;               // --
++studentCount;               // ++
// Relational Operators
boolean isLess = sub1 < sub2;         // <
boolean isLessEqual = sub1 <= sub2;   // <=
boolean isGreater = sub1 > sub2;      // >
boolean isGreaterEqual = sub1 >= sub2;// >=
boolean isEqual = sub1 == sub2;       // ==
boolean isNotEqual = sub1 != sub2;    // !=
// Logical Operators
boolean passedAll = (sub1 >= 35) && (sub2 >= 35);          // &&
boolean topScorer = (sub1 > 90) || (sub2 > 90);            // ||
boolean notFailed = !(sub1 < 35 || sub2 < 35);             // !

System.out.println("\n=== ARITHMETIC OPERATORS ===");
System.out.println("sub1 + sub2        (+)  = " + total);
System.out.println("sub1 - sub2        (-)  = " + difference);
System.out.println("sub1 * sub2        (*)  = " + product);
System.out.println("total / 2          (/)  = " + average);
System.out.println("total % 100        (%)  = " + mod);

System.out.println("\n=== UNARY OPERATORS ===");
System.out.println("Student Count ++         = " + (++studentCount));
```

```
        System.out.println("Student Count --        = " + (--studentCount));
        System.out.println("Student Count ++        = " + (++studentCount));

        System.out.println("\n=== RELATIONAL OPERATORS ===");
        System.out.println("sub1 < sub2        (<)  = " + isLess);
        System.out.println("sub1 <= sub2       (<=) = " + isLessEqual);
        System.out.println("sub1 > sub2        (>)  = " + isGreater);
        System.out.println("sub1 >= sub2       (>=) = " + isGreaterEqual);
        System.out.println("sub1 == sub2       (==) = " + isEqual);
        System.out.println("sub1 != sub2       (!=) = " + isNotEqual);

        System.out.println("\n=== LOGICAL OPERATORS ===");
        System.out.println("(sub1 >= 35 && sub2 >= 35)   (&&) = " + passedAll);
        System.out.println("(sub1 > 90 || sub2 > 90)     (||) = " + topScorer);
        System.out.println("!(sub1 < 35 || sub2 < 35)     (!)  = " + notFailed);

        sc.close();
    }
}
```

Output:

```
D:\JAVA\JAVA_PROGRAMS>java DataOperationsStudentMarksCalculator
Enter marks in Subject 1: 80
Enter marks in Subject 2: 90

=== ARITHMETIC OPERATORS ===
sub1 + sub2        (+)  = 170
sub1 - sub2        (-)  = -10
sub1 * sub2        (*)  = 7200
total / 2          (/)  = 85
total % 100        (%)  = 70

=== UNARY OPERATORS ===
Student Count ++         = 2
Student Count --         = 1
Student Count ++         = 2

=== RELATIONAL OPERATORS ===
sub1 < sub2        (<)  = true
sub1 <= sub2       (<=) = true
sub1 > sub2        (>)  = false
sub1 >= sub2       (>=) = false
sub1 == sub2       (==) = false
sub1 != sub2       (!=) = true

=== LOGICAL OPERATORS ===
(sub1 >= 35 && sub2 >= 35)   (&&) = true
(sub1 > 90 || sub2 > 90)     (||) = false
!(sub1 < 35 || sub2 < 35)     (!)  = true

D:\JAVA\JAVA_PROGRAMS>NEESHNA LAKSHMI_22049
```

Program 3: Data Types

Objective:

This program is designed to demonstrate the usage of all fundamental data types in Java by simulating an employee profile entry system.

Code:

```
import java.util.Scanner;


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


        System.out.print("Enter Employee ID (int): ");
        int empId = sc.nextInt();


        System.out.print("Enter Age (byte): ");
        byte age = sc.nextByte();


        System.out.print("Enter Years of Experience (short): ");
        short exp = sc.nextShort();


        System.out.print("Enter Phone Number (long): ");
        long phone = sc.nextLong();


        System.out.print("Enter Salary (double): ");
        double salary = sc.nextDouble();


        System.out.print("Enter Working Hours per Day (float): ");
        float hours = sc.nextFloat();


        System.out.print("Enter Gender (char - M/F): ");
        char gender = sc.next().charAt(0);
```

```
        System.out.print("Is Permanent Employee? (true/false): ");
        boolean isPermanent = sc.nextBoolean();

        System.out.println("\n===== EMPLOYEE DETAILS =====");
        System.out.println("Employee ID (int): " + empId);
        System.out.println("Age (byte): " + age);
        System.out.println("Experience (short): " + exp + " years");
        System.out.println("Phone Number (long): " + phone);
        System.out.println("Salary (double): ₹" + salary);
        System.out.println("Working Hours (float): " + hours + " hours/day");
        System.out.println("Gender (char): " + gender);
        System.out.println("Permanent Employee (boolean): " + isPermanent);

        sc.close();
    }
}
```

Output:

```
D:\JAVA\JAVA_PROGRAMS>java DataTypesEmployeeRecordEntry
Enter Employee ID (int): 22049
Enter Age (byte): 28
Enter Years of Experience (short): 5
Enter Phone Number (long): 9876543210
Enter Salary (double): 65000.50
Enter Working Hours per Day (float): 8.5
Enter Gender (char - M/F): F
Is Permanent Employee? (true/false): true

===== EMPLOYEE DETAILS =====
Employee ID (int): 22049
Age (byte): 28
Experience (short): 5 years
Phone Number (long): 9876543210
Salary (double): ?65000.5
Working Hours (float): 8.5 hours/day
Gender (char): F
Permanent Employee (boolean): true

D:\JAVA\JAVA_PROGRAMS>
```

Program 4: CONTROL STATEMENTS

Objective:

The aim of this program is to demonstrate the use of all basic control statements in Java including if, if-else, else-if, switch, for, while, and do-while. The program simulates a car rental booking system to validate user age, calculate rental cost, log travel activity, apply discounts, and confirm the booking.

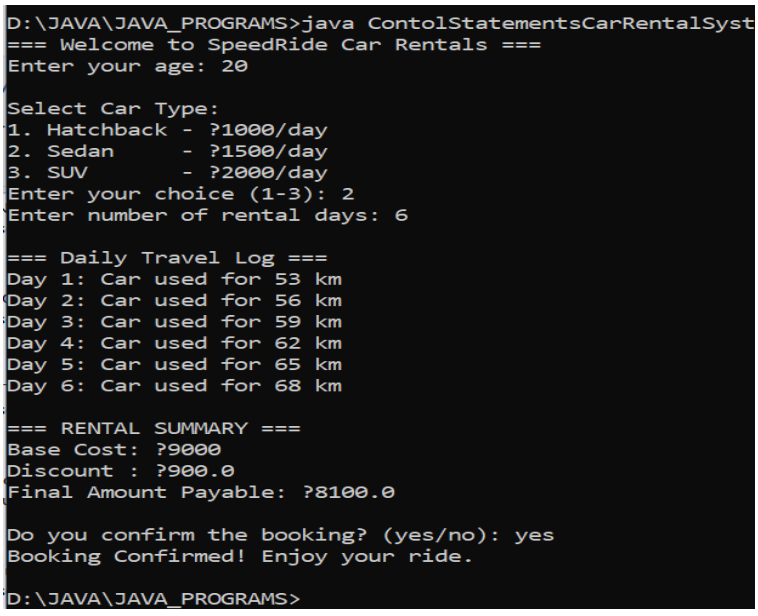
Code:

```
import java.util.Scanner;


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


        System.out.println("=== Welcome to SpeedRide Car Rentals ===");


        // Age validation using while loop
        System.out.print("Enter your age: ");
        int age = sc.nextInt();
        while (age < 18) {
            System.out.print("You must be at least 18 to rent a car. Re-enter age: ");
            age = sc.nextInt();
        }


        // Car selection using switch
        System.out.println("\nSelect Car Type:");
        System.out.println("1. Hatchback - ₹1000/day");
        System.out.println("2. Sedan     - ₹1500/day");
        System.out.println("3. SUV       - ₹2000/day");
        System.out.print("Enter your choice (1-3): ");
        int carType = sc.nextInt();
```

```
int dailyRate = 0;
switch (carType) {
    case 1: dailyRate = 1000; break;
    case 2: dailyRate = 1500; break;
    case 3: dailyRate = 2000; break;
    default: System.out.println("Invalid car type selected. Defaulting to Hatchback.");
            dailyRate = 1000;
}
// Rental duration
System.out.print("Enter number of rental days: ");
int days = sc.nextInt();
int baseCost = dailyRate * days;

// Travel logs using for loop
System.out.println("\n=== Daily Travel Log ===");
for (int i = 1; i <= days; i++) {
    System.out.println("Day " + i + ": Car used for " + (50 + i * 3) + " km");
}
// Discount offers using if-else-if
double discount = 0;
if (days >= 10) {
    discount = 0.20 * baseCost;
} else if (days >= 5) {
    discount = 0.10 * baseCost;
} else if (days >= 3) {
    discount = 0.05 * baseCost;
}
double finalAmount = baseCost - discount;

// Display invoice
```

```
        System.out.println("\n=== RENTAL SUMMARY ===");
        System.out.println("Base Cost: ₹" + baseCost);
        System.out.println("Discount : ₹" + discount);
        System.out.println("Final Amount Payable: ₹" + finalAmount);
        // Confirmation using do-while
        String confirm;
        do {
            System.out.print("\nDo you confirm the booking? (yes/no): ");
            confirm = sc.next().toLowerCase();
        } while (!confirm.equals("yes") && !confirm.equals("no"));

        if (confirm.equals("yes")) {
            System.out.println("Booking Confirmed! Enjoy your ride.");
        } else {
            System.out.println("Booking Cancelled.");
        }
        sc.close();
    }
}
```

Output:



Program 4: ACCESS MODIFIERS

Objective:

Java Program to Demonstrate Access Modifiers with Abstract Class and Inheritance.

Code:

```
// Abstract class - cannot be instantiated directly
abstract class BankAccount {
    private long accountNumber;         // private
    private double balance;             // private
    protected String accountHolder;     // protected
    final double MIN_BALANCE = 500.0;   // final

    // Constructor (default access)
    BankAccount(long accNo, String name, double initialDeposit) {
        this.accountNumber = accNo;
        this.accountHolder = name;
        this.balance = initialDeposit;
    }

    // Public method to deposit money
    public void deposit(double amount) {
        balance += amount;
        System.out.println("Deposited ₹" + amount);
    }

    // Public method to show balance
    public void showBalance() {
        System.out.println("Balance for " + accountHolder + ": ₹" + balance);
    }

    // Protected method for internal audit
```

```
    protected void internalAudit() {
        System.out.println("Account " + accountNumber + " audited. Balance: ₹" + balance);
    }

    // Abstract method to be implemented by child class
    public abstract void withdraw(double amount);
}

// Subclass
class SavingsAccount extends BankAccount {

    // Constructor
    public SavingsAccount(long accNo, String name, double initialDeposit) {
        super(accNo, name, initialDeposit);
    }

    // Implement abstract method
    public void withdraw(double amount) {
        if (amount > 0 && amount <= 10000) {
            System.out.println(accountHolder + " withdrawing ₹" + amount + "...");
        } else {
            System.out.println("Invalid withdrawal amount or limit exceeded.");
        }
    }
    // Default method (no modifier)
    void printAccountType() {
        System.out.println("Account Type: Savings Account");
    }
}
```

```
// Main class
public class AccessModifiersBankSystemDemo {
    public static void main(String[] args) {
        SavingsAccount acc = new SavingsAccount(1234567890L, "Ananya Sharma", 5000.0);

        acc.printAccountType();      // default access
        acc.deposit(2500.0);         // public method
        acc.withdraw(3000.0);        // public method from abstract
        acc.showBalance();           // public method

        // acc.accountNumber ✕ private, not accessible
        // acc.balance       ✕ private, not accessible

        acc.internalAudit();         // protected method
    }
}
```

Output:

```
D:\JAVA\JAVA_PROGRAMS>java AccessModifiersBankSystemDemo.java
Account Type: Savings Account
Deposited ?2500.0
Ananya Sharma withdrawing ?3000.0...
Balance for Ananya Sharma: ?7500.0
Account 1234567890 audited. Balance: ?7500.0

D:\JAVA\JAVA_PROGRAMS>
```

&&&&&&&&&&&&&