Lab Sheet: Java Programming Fundamentals

Course: Introduction to Programming Language II (Java)
Topic: Input/Output, Conditionals, Loops, Arrays, Methods & Basic Classes
Level: Beginner
Duration: 2–2.5 hours

Prepared by: Ashraful Islam Paran, Dept. of CSE, Southeast University

1. Objectives

By the end of this lab, students will be able to:

- Read different types of input using `Scanner`
- Use `if-else`, `switch`, and nested conditions
- Work with `for`, `while`, and `do-while` loops
- Declare, initialize and traverse arrays
- Create and call simple methods (with & without return values)
- Understand basic class-object creation (non-OOP heavy)
- Solve simple algorithmic problems using the above concepts

2. Theoretical Background

Java programs usually follow this flow:

- Input → using `Scanner`
- Decision making → `if`, `else if`, `switch`
- Repetition → `for`, `while`, `do-while`
- Data collection → arrays
- Code organization → methods

Basic input pattern:

```
import java.util.Scanner;
Scanner sc = new Scanner(System.in);
int n = sc.nextInt();
double d = sc.nextDouble();
String line = sc.nextLine();
```

3. Example 1: Basic Input – Integer, Double, String

```
import java.util.Scanner;

public class Main {
    public static void main(String args[]) {
        Scanner scanner = new Scanner(System.in);
        int n = scanner.nextInt();
        double cg = scanner.nextDouble();
        System.out.println(n);
        scanner.nextLine();            // consume newline
        String x = scanner.nextLine();
        System.out.println(x);
    }
}
```

4. Example 2: If-Else – Adult or Minor

```
import java.util.Scanner;

public class IfElseExample {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter your age: ");
        int age = sc.nextInt();

        if (age >= 18) {
            System.out.println("You are an adult.");
        } else {
            System.out.println("You are a minor.");
        }
    }
}
```

5. Example 3: Even-Odd Checker

```
import java.util.Scanner;

public class EvenOdd {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int num = sc.nextInt();

        if (num % 2 == 0)
            System.out.println(num + " is even.");
        else
            System.out.println(num + " is odd.");
    }
}
```

6. Example 4: Switch – Day Name (1–3)

```
import java.util.Scanner;

public class SwitchExample {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a day number (1-3): ");
        int day = sc.nextInt();

        switch (day) {
            case 1: System.out.println("Monday"); break;
            case 2: System.out.println("Tuesday"); break;
            case 3: System.out.println("Wednesday"); break;
            default: System.out.println("Invalid day");
        }
    }
}
```

7. Example 5: Loops – for, while, do-while

```
// for loop
for (int i = 1; i <= 5; i++) {
    System.out.println("Number: " + i);
}

// while loop
int i = 1;
while (i <= 5) {
    System.out.println("Count: " + i);
    i++;
}

// do-while loop
int j = 1;
do {
    System.out.println("Value: " + j);
    j++;
} while (j <= 5);
```

8. Example 6: Simple Array Declaration & Traversal

```
public class ArraysExample {
    public static void main(String[] args) {
        int[] numbers = {10, 20, 30, 40};

        // int[] numbers = new int[]{1, 2, 3, 4, 5};

        // int[] numbers = new int[5];

        // numbers = new int[]{1, 2, 3, 4, 5};

        for (int i = 0; i < numbers.length; i++) {
```

```
            System.out.println("Element " + i + ": " + numbers[i]);
        }
    }
}
```

9. Example 7: Read Array Size Elements from User

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int[] x = new int[n];

        for (int i = 0; i < n; i++) {
            x[i] = sc.nextInt();
        }

        // You can now print or process x[]
    }
}
```

10. Example 8: Simple Method – Addition

```
public class Main {
    static int add(int a, int b) {
        return a + b;
    }

    public static void main(String[] args) {
        int result = add(5, 7);
        System.out.println("Sum: " + result);
    }
}
```

11. Example 9: Factorial using Loop

```
public class Factorial {
    public static void main(String[] args) {
        int n = 5;
        int fact = 1;

        for (int i = 1; i <= n; i++) {
            fact *= i;
        }

        System.out.println("Factorial of " + n + " = " + fact);
    }
}
```

12. Example 10: Reverse a String

```java
public class ReverseString {
    public static void main(String[] args) {
        String str = "Java";
        String rev = "";

        for (int i = str.length() - 1; i >= 0; i--) {
            rev += str.charAt(i);
        }

        System.out.println("Reversed: " + rev);
    }
}
```

13. Example 11: Very Basic Class & Object

```java
class Student {
    String name;
    int id;

    void display() {
        System.out.println("Name: " + name + ", ID: " + id);
    }
}

public class ClassObjectExample {
    public static void main(String[] args) {
        Student s1 = new Student();
        s1.name = "Jim";
        s1.id = 101;
        s1.display();
    }
}
```

14. Practice Tasks

1. Write a program that reads two integers and prints the larger one.
2. Create a program that reads a month number (1–12) and prints the month name using `switch`.
3. Read 10 integers into an array and print their sum and average.
4. Write a method `isPrime(int n)` that returns `true` if n is prime.
5. Read a string from user and count how many vowels it contains.
6. Create a simple `Book` class with fields `title`, `author`, `price` and a `display()` method.
7. Write a program that reads n numbers and finds the maximum and minimum value.

Lab Tasks

Task 1 — Sum of Array Elements. Take `n` numbers in an array from the user and print their sum.

Example Input:

```
Enter array size: 5
Enter 5 numbers: 10 20 30 40 50
```

Example Output:

```
Sum = 150
```

Task 2 — Find the Largest Number. Take `n` integers in an array from the user and print the largest element.

Example Input:

```
Enter array size: 5
Enter 5 numbers: 10 55 22 14 9
```

Example Output:

```
Largest = 55
```

Task 3 — Count Even and Odd Numbers. Count how many numbers are even and how many are odd in an array input by the user.

Example Input:

```
Enter array size: 6
Enter 6 numbers: 1 2 3 4 5 6
```

Example Output:

```
Even: 3
Odd: 3
```

Task 4 — Reverse an Array. Take `n` numbers from the user, store them in an array, and print them in reverse order.

Example Input:

```
Enter array size: 5
Enter 5 numbers: 10 20 30 40 50
```

Example Output:

```
Original array: 10 20 30 40 50
Reversed array: 50 40 30 20 10
```

Task 5 — Search an Element. Take `n` numbers in an array from the user, ask for a number to search, and show whether it is found.

Example Input:

```
Enter array size: 5
Enter 5 numbers: 2 4 6 8 10
Enter number to search: 6
```

Example Output:

```
6 found at index 2
```

Task 6 — Average of Array Elements. Input `n` numbers into an array from the user and calculate their average.

Example Input:

```
Enter array size: 5
Enter 5 numbers: 10 20 30 40 50
```

Example Output:

```
Average = 30.0
```

Task 7 — Copy One Array to Another. Input `n` numbers into an array from the user, copy them into another array, and display both arrays.

Example Input:

```
Enter array size: 5
Enter 5 numbers: 1 2 3 4 5
```

Example Output:

```
Original array: 1 2 3 4 5
Copied array:   1 2 3 4 5
```

Task 8 — Smallest and Largest. Input `n` numbers from the user and find both smallest and largest numbers in the array.

Example Input:

```
Enter array size: 5
Enter 5 numbers: 3 9 1 7 5
```

Example Output:

```
Smallest = 1
Largest = 9
```

15. Quick Reference Table

Concept	Keyword/Structure	Common Use
Input	`Scanner`	Reading numbers, strings
Decision	`if-else`, `switch`	Conditions, menu
Repetition	`for`, `while`, `do-while`	Loops, counting
Collection	`int[]`, `String[]`	Lists of data
Function	`static` method	Reusable code
Basic OOP	`class`, `new`	Simple objects