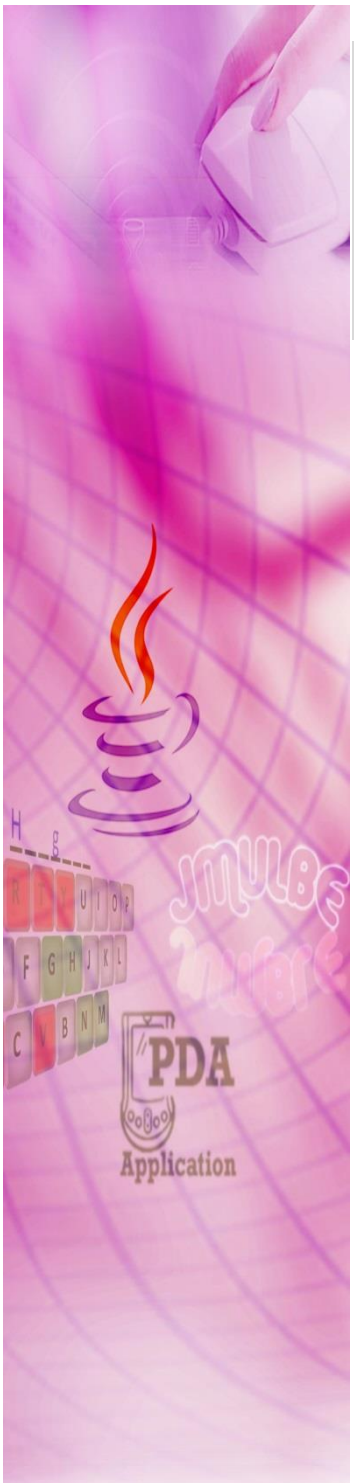


# Instructor Inputs

Session 4

# Session Overview

This session covers Chapter 3 of the book, Introduction to Java – SG. This session will help the students to become familiar with the various conditional constructs in Java. In addition, the students will learn how to implement the different types of looping constructs.

## Working with Conditional Constructs

### Handling Tips

Start the discussion by asking the students how they take decisions in their daily life. Elicit the responses, and then start explaining the students that decision making in real life is important. Thereafter, inform the students that the decision making technique can be incorporated in the Java programs by using the conditional constructs. Further, explain that Java supports different types of conditional constructs. Thereafter, explain each conditional construct with the help of the example of the Classic Jumble Word game. While discussing the `if` construct, you can explain the students that the result of the condition specified in the `if` construct will be reflected only on the single immediate next statement or on a block of statements enclosed within the braces. Next, discuss the following code snippet with the previous explanation:

```
int a=3, b=20;

if(a>b)
  System.out.println("A is greater");
  System.out.println("B is greater");
```

The preceding code snippet will display the following output:

```
B is greater
```

Further, explain the students that in the preceding code snippet, the first print statement is skipped and the second print statement is executed because the condition is false. The impact of the condition is reflected only on the first print statement.

Next, while discussing the `switch` construct, highlight the point that the `default` statement and the `case` statements can occur in any order in the `switch` construct. In addition, tell the students that a single statement can carry multiple case constants with the example given in the Additional Examples topic. Thereafter, inform the students that if the `break` statement is not specified within the case constants, all the subsequent case constants will get executed.

### Additional Examples

The following code snippet demonstrates the example of a single statement carrying multiple case constants:

```
char ch = 'i';
switch(ch)
{
    case 'a' :
    case 'e' :
    case 'i' :
```

```
        case 'o' :
        case 'u' : System.out.println("It is a vowel");
                   break;
        default: System.out.println("It is not a vowel");
    }
```

The preceding code snippet will display the following output:

```
It is a vowel
```

## Activity 3.1: Working with Conditional Constructs

### Handling Tips

Discuss the problem statement with the students.

To perform the activity, 3.1, you need to use the **Activity1.1.txt** file, which is provided at the following location in the TIRM CD:

- **Datafiles For Faculty\Activities\Chapter 03\Activity 3.1\Input Files**

The solution file, **Hangman.java**, for this activity is provided at the following location in the TIRM CD:

- **Datafiles For Faculty\Activities\Chapter 03\Activity 3.1\Solution**

## Working with Loop Constructs

Start the discussion by explaining the students why looping constructs are important and how they can be implemented in Java. Further, explain the types of looping constructs available in Java. Thereafter, explain each construct with the help of the example of the Classic Jumble Word game.

Next, while discussing the `for` loop, you can discuss the example given in the Additional Examples topic. Thereafter, inform the students that Java also provides another type of loop, `for-each` loop, which is used to iterate the values of the array. However, inform the students that the `for-each` loop will be covered in the later chapters.

Further, explain the nesting of loop construct with the help of the information given in the Additional Inputs topic.

### Additional Inputs

Apart from the nesting of conditional construct, Java also supports the nesting of loop constructs. The nesting of loop constructs can be achieved with the same loop construct, as well as with different loop constructs.

The following code snippet shows the nesting of the `while` loop within the `for` loop construct:

```
for(int i=0;i<3;i++)
{
    System.out.println("The value of i is "+i);
    int j=0;
    while(j<20)
    {
        System.out.println("j "+j);
```

```
            j+=5;
        }
    }
```

The preceding code snippet will display the following output:

```
The value of i is 0
j 0
j 5
j 10
j 15
The value of i is 1
j 0
j 5
j 10
j 15
The value of i is 2
j 0
j 5
j 10
j 15
```

### Additional Examples

The following code snippet demonstrates the example of more than one loop combined into a single loop with the single condition:

```
for(int a=0,b=10;b<50;a++,b+=10)
{
    System.out.println("a: "+a);
    System.out.println("b: "+b);
}
```

The preceding code snippet will display the following output:

```
a: 0
b: 10
a: 1
b: 20
a: 2
b: 30
a: 3
b: 40
```

## Activity 3.2: Working with Loop Constructs

### Handling Tips

Discuss the problem statement with the students.

To perform the activity, 3.2, you need to use the **Activity3.1.txt** file, which is provided at the following location in the TIRM CD:

- **Datafiles For Faculty\Activities\Chapter 03\Activity 3.2\Input Files**

The solution file, **Hangman.java**, for this activity is provided at the following location in the TIRM CD:

- **Datafiles For Faculty\Activities\Chapter 03\Activity 3.2\Solution**

## FAQs

- *Can the `break` and `continue` statements be used together inside the `for` loop?*

  Ans: Yes, the `break` and `continue` statements can be used together inside the `for` loop. For this, you need to define the conditions separately for the `break` statement, as well as for `continue` statement.

- *Can a `switch` construct be created inside another `switch` construct?*

  Ans: Yes, a `switch` construct can be created inside another `switch` construct. This is called the nested `switch` construct. Each `switch` construct creates its own block of `case` constants. Therefore, no conflict occurs between the `case` constants of the inner and outer `switch` constructs.

- *Can the conditional construct be used inside the loop construct and vice-versa?*

  Ans: Yes, the conditional construct can be used inside the loop construct and vice-versa.