

Java - Introduction to Programming

Lecture 3

Conditional Statements

The if block is used to specify the code to be executed if the condition specified in if is true, the else block is executed otherwise.

If statement

```
if (condition)
{
    //Statements to be executed
}
```

```
public class ifelse {
    public static void main(String[] args) {
double a = 0.8;
    if (a>0)
    {
        System.out.println(a + " is a Positive Number!");
    }
}
}
```

If..else statement

```
if (condition)
{
    //Statements to be executed if condition satisfies
}
else
{
    //Statements to be executed if the condition is not satisfied
}
```

```
public class ifelse {
    public static void main(String[] args) {
        double a = -0.8;
        if (a>0)
        {
            System.out.println(a + " is a Positive Number!");
        }
        else
        {
            System.out.println(a + " is a Negative Number!");
        }
    }
}
```

```
        }
    }
}
```

nested if()...else

- The nested if()...else statements check all the inner conditional statements once the outer conditional if() statement is satisfied.
- Nested if()...else statements take more execution time (they are slower) in comparison to an else if()... ladder.

```
if (condition1) {

    // Statements to execute if condition1 is true

    if (condition2) {

        // Statements to execute if condition1 and condition2 are both true

    } else {

        // Statements to execute if condition1 is true and condition2 is false

    }

} else {

    // Statements to execute if condition1 is false

}

}
```

```
public class ifelse {
    public static void main(String[] args) {
        int a = 2;
        int b = 2;
        int c = 2;

        if (a == b) {

            // nested if else condition to check if c is equal to a and b
            if (a == c) {
                // all are equal
                System.out.println("Equal");
            } else {
                // a!=c
                System.out.println("Not equal");
            }
        } else {
            // a!=b
            System.out.println("Not equal");
        }
    }
}
```

```
    }  
}
```

if-else-if ladder

- A linear sequence of conditions, where only one condition can be true at a time.
- Nested if()...else statements take more execution time (they are slower) in comparison to an else if()... ladder.

```
if (condition)  
{  
    //Statements set 1  
}  
else if (condition 2)  
{  
    //Statements set 2  
}  
...  
else  
{  
    //Statements to be executed if no condition is satisfied.  
}
```

```
public class ifelse {  
    public static void main(String[] args) {  
        // initializing expression  
        int i = 20;  
  
        // condition 1  
        if (i >= 10)  
            System.out.println("i is 10\n");  
  
        // condition 2  
        else if (i < 15)  
            System.out.println("i is 15\n");  
  
        // condition 3  
        else if (i == 20)  
            System.out.println("i is 20\n");  
  
        else  
            System.out.println("i is not present\n");  
  
        System.out.println("Outside if-else-if");  
    }  
}
```

Quiz:

If 3 is even then print “Hello”

```
1. if(3%2==0){  
    System.out.println("Hello");  
}  
1. if(3%2==0)  
    System.out.println("Hello");  
2. if(3%2==0);  
    System.out.println("Hello");
```

2. Conditional Statements ‘switch’

Switch case statements are a substitute for long if statements that compare a variable to multiple values. After a match is found, it executes the corresponding code of that value case.

The following example is to print days of the week:

```
//print days of the week  
import java.util.*;  
public class printdays {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        int n = sc.nextInt();  
        switch(n) {  
            case 1 :  
                System.out.println("Monday");  
                break;  
            case 2 :  
                System.out.println("Tuesday");  
                break;  
            case 3 :  
                System.out.println("Wednesday");  
                break;  
            case 4 :  
                System.out.println("Thursday");  
                break;  
            case 5:  
                System.out.println("Friday");  
                break;  
            case 6 :  
                System.out.println("Saturday");  
                break;  
            default :  
                System.out.println("Invalid Entry");  
        }  
    }  
}
```

```
}
```

Loops

A loop is used for executing a block of statements repeatedly until a particular condition is satisfied. A loop consists of an initialization statement, a test condition and an increment statement.

For Loop

The syntax of the for loop is :

```
for (initialization; condition; update) {  
    // body of-loop  
}
```

Initialization: is executed (one time) before the execution of the code block.

Condition: defines the condition for executing the code block.

Update: is executed (every time) after the code block has been executed.

```
public class loop {  
    public static void main(String[] args) {  
        //for loop  
        for(int i =1; i<=10;i++){  
            System.out.println(i);  
        }  
    }  
}
```

Nested Loop

- place a loop inside another loop. This is called a **nested loop**.

```
• public class ifelse {  
•     public static void main(String[] args) {  
•         // Outer loop  
•         for (int i = 1; i <= 2; i++) {  
•             System.out.println("Outer: " + i); // Executes 2 times  
•         }  
•         // Inner loop  
•         for (int j = 1; j <= 3; j++) {  
•             System.out.println(" Inner: " + j); // Executes 6 times (2 * 3)  
•         }  
•     }  
• }
```

For-Each Loop

There is also a "for-each" loop, which is used exclusively to loop through elements in an array (or other data sets):

```
for (type variableName : arrayName) {  
    // code block to be executed  
}
```

```
public class ifelse {  
    public static void main(String[] args) {  
        String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};  
        for (String i : cars) {  
            System.out.println(i);  
        }  
    }  
}
```

While Loop:

- The while loop loops through a block of code as long as a specified condition is true:

The syntax for while loop is :

```
while(condition) {  
    // body of the loop  
}
```

```
while loop  
int i=0;  
while (i<11) {  
    System.out.println(i);  
    i++;  
}
```

Do-While Loop

The syntax for the do-while loop is :

```
do {  
    // body of loop;  
}  
while (condition);
```

```
// do while  
int i =11;  
do{  
    System.out.println(i);  
    i++;
```

```
}while(i<11);
```

Print sum of n natural numbers

```
//print sum of n natural numbers
import java.util.*;
public class natural {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("enter number:");
        int n = sc.nextInt();
        int sum = 0;
        for(int i=0; i<=n; i++){
            sum = sum+i;
        }
        System.out.println(sum);
    }
}
```

Home Assignment:

- Create a program to test two values to find out if 20 is greater than 18. If the condition is true, print some text (using if statement)
- Create a program to test two values to find out if 20 is greater than 18. If the condition is true, print some text, If the condition is false, print some text (using if else statement)
- Make a Calculator. Take 2 numbers (a & b) from the user and an operation as follows :
 - + (Addition) a + b
 - - (Subtraction) a – b
 - * (Multiplication) a * b
 - / (Division) a / b
 - % (Modulo or remainder) a % bCalculate the result according to the operation given and display it to the user. (Using if else ladder)
- Ask the user to enter the number of the month & print the name of the month. For eg - For '1' print 'January', '2' print 'February' & so on. (Using switch)
- create a program that only print even values between 0 and 10 (using for loop)
- Write a Program to find maximum in an array (Using for each loop)
- WAP to Calculating the Sum of Numbers from 1 to 10 (while Loop)

- WAP to Calculating the Sum of Numbers from 1 to 10 (do while Loop)