JPL :: While ... Loops

TalentSprint

Licensed To Skill

Version 1.0.4

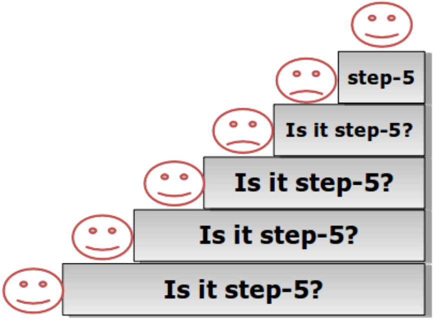
Learning Objectives

By the end of this presentation, you will be able to:

- Learn the concept of looping in solutions to problems
- Use while loop and do...while loop in Java code.

While...Loops

The smiley wants to reach step-5 from bottom



step-5

Is it step-5?

Is it step-5?

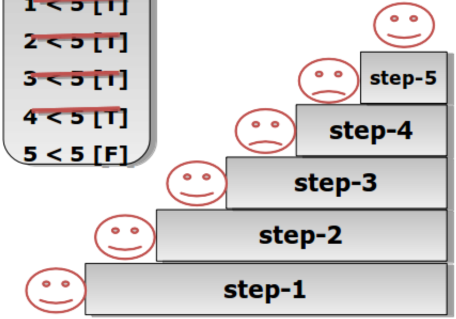
Is it step-5?

Is it step-5?

While...Loops

```
step = 0
if(step < 5)
  step +=1
if(step < 5)
  step +=1
if(step < 5)
  step +=1
if(step < 5)
  step +=1
if(step < 5)
  step +=1
if(step < 5)
```

Step	Condition
~~0~~	~~0 < 5 [T]~~
~~1~~	~~1 < 5 [T]~~
~~2~~	~~2 < 5 [T]~~
~~3~~	~~3 < 5 [T]~~
~~4~~	~~4 < 5 [T]~~
5	5 < 5 [F]



Reache

Can you see some pattern in this process?

While...Loops

```
step = 0
while(step<5)
{
    step +=1
}
if(step < 5)
    step +=1
if(step < 5)
    step +=1
if(step < 5)
    step +=1
if(step < 5)
    step +=1
```

Reached

Step	Condition
~~0~~	~~0 < 5 [T]~~
~~1~~	~~1 < 5 [T]~~
~~2~~	~~2 < 5 [T]~~
~~3~~	~~3 < 5 [T]~~
~~4~~	~~4 < 5 [T]~~
5	5 < 5 [F]

nment and comparing

PEATEDLY.

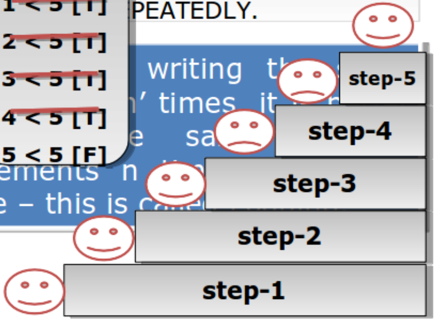
writing th

' times it

e sa

statements n

once – this is c



While...Loops

To find largest of ’n’ numbers

4, 9, 2, -27, 34, 26, 45, 64, 58, 12, 96, ..., n

↑

Now, which is the largest number?

How about the following approach:

```
Read first number. Call it largest_so_far.
Read second number into next.
if (next > largest_so_far) largest_so_far = next
.....
Read ’n’th number into next.
if (next > largest_so_far) largest_so_far = next
print largest_so_far
```

What if n is a big number !!!

While...Loops

```
largest = Integer.parseInt(args[0]);

next = Integer.parseInt(args[1]);

while(count < args.length){
    if (next > largest)
        largest = next
    next = Integer.parseInt(args[count]);
    count++;
}
```

While...Loops

Method 1: Java Code

```
public class LargestNumberMany {
    public static void main(String[] args) {
        int next, largestSoFar, count = 0;
        largestSoFar = Integer.parseInt(args[count]);
        count ++;
        while (count < args.length) {
            next = Integer.parseInt(args[count]);
            count ++;
            if  (next > largestSoFar)
                largestSoFar = next;
        }
        System.out.println("Largest: " + largestSoFar);
    }
}
```

While...Loops

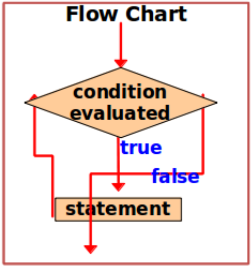
‘While’ Statement

The ‘**while**’ statement is a control flow looping statement that allows the code to be executed repeatedly based on a given Boolean condition. The ‘**while**’ statement can be thought of as a repeating ‘**if**’ statement.

While...Loops

Syntax

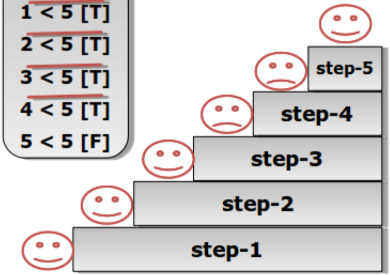
```
while (expression) {
        statement(s);
}
```



While...Loops

```
do {
      step += 1;
} while (step < 5);
```

Step	Condition
~~0~~	
~~1~~	~~1 < 5 [T]~~
~~2~~	~~2 < 5 [T]~~
~~3~~	~~3 < 5 [T]~~
~~4~~	~~4 < 5 [T]~~
5	5 < 5 [F]



While...Loops

Method 2: Java Code

```
public class LargestNumberMany2 {
    public static void main(String[] args) {
        int next, largestSoFar, count = 0;
        largestSoFar = Integer.parseInt(args[count++]);
        do {
            next = Integer.parseInt ( args[count++] );
            if  (next > largestSoFar)
                largestSoFar = next;
        }  while (count < args.length);
        System.out.println("Largest: " + largestSoFar);
    }
}
```

While...Loops

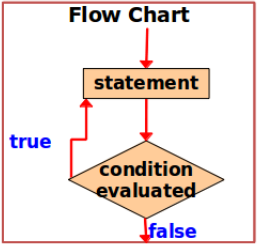
‘**do** – **while**’ Statement

The purpose of ‘**do** – **while**’ statement is the same as that of ‘**while**’ statement. However, unlike ‘**while**’ statement, ‘**do** – **while**’ evaluates its expression at the bottom of the loop. Therefore, the statements within the ‘**do**’ block are always executed at least once.

While...Loops

Syntax

```
do {
      statement(s);
} while (expression);
```



While...Loops

1. Write Java code using **while** loop and **do... while** loop for sum of numbers problem.
2. Write a program to find out whether a given number is a perfect square or not.
3. Write a program to find reverse of a given number
4. Write a program to find sum of digits problem
5. Write a program to find given number is perfect number or not.

While...Loops

Solution - Exercise 1 (Method 1)

```
public class SumMany {
    public static void main(String[] args) {
        int next, sumSoFar = 0, count = 0;
        while (count < args.length) {
            next = Integer.parseInt(args[count]);
            count ++;
            sumSoFar += next;
        }
        System.out.println("Sum: " + sumSoFar);
    }
}
```

While...Loops

Solution - Exercise 1 (Method 2)

```
public class SumMany2 {
    public static void main(String[] args) {
        int next, sumSoFar = 0, count = 0;
        do {
            next = Integer.parseInt(args[count]);
            count ++;
            sumSoFar += next;
        } while (count < args.length);
        System.out.println("Sum: " + sumSoFar);
    }
}
```

While...Loops

Solution - Exercise 2

```
public class PerfSquare {
    public static void main(String[] args) {
        int i = 1;
        int givenNumber = Integer.parseInt(args[0]);
        while (i < givenNumber) {
            if (i * i == givenNumber) {
                System.out.println(givenNumber + " is perfect
    square");
                return;
            }
            i++;
        }
        System.out.println(givenNumber + " is not perfect
    square");
    }
}
```

While...Loops