JPL :: For...Loops

TalentSprint

Licensed To Skill

Version 1.0.4

Learning Objectives

The content in this presentation is aimed at teaching learners to:

- Write solutions for simple problems needing iterative execution of a set of steps a certain number of times.
- Write Java programs using **for** loops

For...Loops

Sum of 'n' Numbers

10, 7, 232, 44, 8, 526, 49, 64, 98, 12, ..., n

⇑

What is the sum?

How about the following approach:

Read number of numbers (n).
Repeat 'n' times.
Read next number (new_number)
sum_so_far = sum_so_far + new_number
Print sum_so_far

For...Loops

Solution for finding Sum of 'n' Numbers

Read n;

sumSoFar = 0;

for i = 1 to n

Read next number (newNumber)

sumSoFar = sumSoFar + newNumber

print sumSoFar

For...Loops

Java Code to find Sum of 'n' Numbers Using - 'for' Loop

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

For...Loops

java SumWithFor 2 3 4 2

Output

Sum: 11

Before entering loop, sumSoFar = 0

count	**sumSoFar**	**condition**
0	0	T
1	2	T
2	5	T
3	9	T
4	11	F

For...Loops

'**for**' Statement

The '**for**' statement allows you to repeat execution of a set of statements a specific number of times.

Syntax of 'for' Statement:

```
for (initialization ; termination; increment/decrement) {
    statement(s);
}
```

For...Loops

Write Java code, using '**for**' loop, to find largest number among 'n' numbers.

For...Loops

Solution:

```
public class LargestWithFor {
    public static void main(String[] args) {
        int next, count, largestSoFar;
        for (count=1; count < args.length; count++) {
            next = Integer.parseInt(args[count]);
        }
        System.out.println("Largest: " + largestSoFar);
    }
}
```

For...Loops

Find Even or Odd numbers among the numbers from 1 to ‘n’

Solution:

```
Read n
for x from 1 to n
      if  (x % 2 == 0)
            printf("%d", x , ":  is  even.");
      else
            printf("%d", x, ":  is  odd.");
```

For...Loops

Java Code to Find Odd or Even Numbers:

```
public class EvenNumber {
    public static void main(String[] args) {
        int i;
        int givenNumber = Integer.parseInt(args[0]);
        for (i = 1; i <= givenNumber; i++) {
            if (i % 2 == 0)
                System.out.println(i + " is even.");
            else
                System.out.println(i + " is odd.");
        }
    }
}
```

For...Loops

The Problem

Find if a number is a Prime number or not.

High Level Solution

Check if there is a divisor to given number which is other than 1 and itself. If there is, the number is not prime. Otherwise, prime.

For...Loops

Detailed Solution

Read the number into n.
for i from 2 to n−1,
    if n % i = 0, then print ("n not prime").
Print ("n prime");

For...Loops

Java Code to Find Prime Numbers:

```
public class PrimeNumber {
    public static void main(String[] args) {
        int i;
        int givenNumber = Integer.parseInt(args[0]);
        for (i = 2; i <= givenNumber – 1; i++) {
            if (givenNumber % i == 0) {
                System.out.println (givenNumber + " is not
    prime.");
                return;
            }
        }
        System.out.println(givenNumber + " is prime");
    }
}
```

For...Loops

Write Java code for printing first ‘n’ odd numbers.

For...Loops