File Edit View Help

File Edit View Help

Loops
Part 02

Looping Statements

- For-statement
- Counting Loop
- Special kind of While-Statement
- Arguments require 3 parts, separated by semicolons
  - Init: This initialization of a counting variable. Only runs once.
  - Boolean Expression: Just like before
  - Update: Updates the counting variable after all other statements in the body have run
- Putting curly braces “{}” to denote the body of the for-statement is strongly encouraged
- Do not put a semicolon “;” after the parenthesis
- Spoken
  - “Do that for this many times”

Syntax

```
for(<<Init>> ; <<Boolean expression>> ; <<Update>>)
{
      //Body of the for-statement
}
//Outside Body of the for-statement
```

Examples

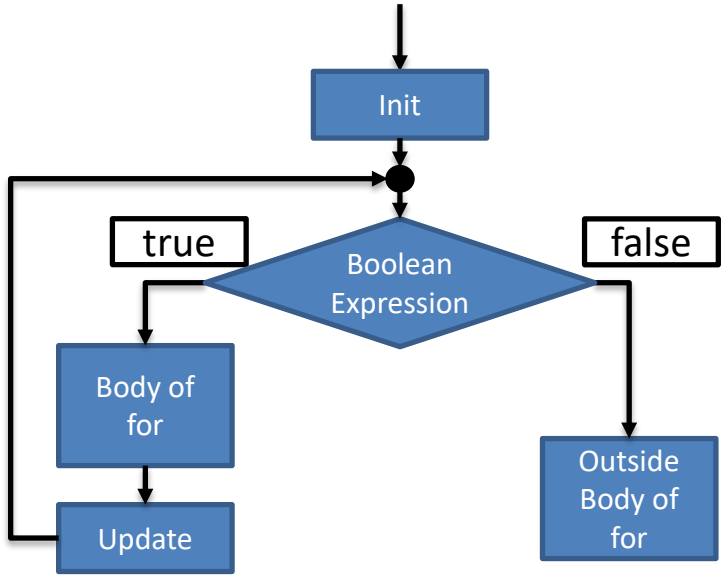
```
for(int i=0; i<10; i++)
{
      System.out.println(i);
}
```

Looping Statements

Syntax

```
for(<<Init>> ; <<Boolean expression>> ; <<Update>>)
{
	//Body of the for-statement
}
//Outside Body of the for-statement
```

General For-Statement Flow Chart



Looping Statements

For-Loop

```
for(<<Init>> ; <<Boolean expression>> ; <<Update>>)
{
      //Body of the for-statement
}
//Outside Body of the for-statement
```

While-Loop

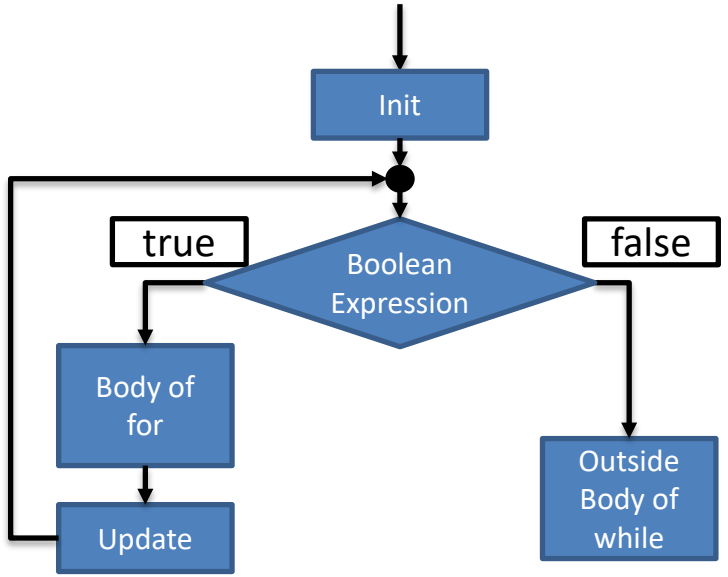
```
<<Init>>;
while(<<Boolean expression>>)
{
      //Body of the while-statement
       <<Update>>;
}
//Outside Body of the while-statement
```

Looping Statements

Syntax

```
for(<<Init>> ; <<Boolean expression>> ; <<Update>>)
{
	//Body of the for-statement
}
//Outside Body of the for-statement
```

General For-Statement Flow Chart

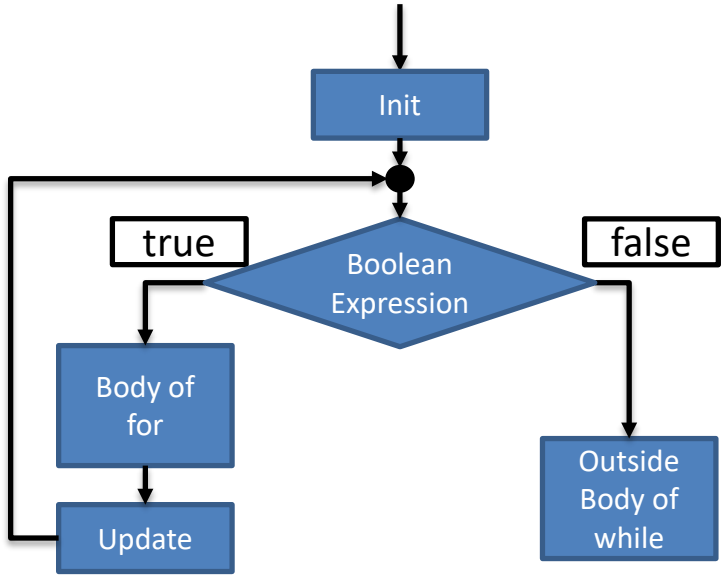


Looping Statements

Syntax

```
<<Init>>;
while(<<Boolean expression>>)
{
       //Body of the while-statement
        <<Update>>;
}
//Outside Body of the while-statement
```

General For-Statement Flow Chart



File Edit View Help

Example

File Edit View Help

Nested Example

File Edit View Help

Closer Look

```
for(int i=0;i<length;i++)
{
    for(int j=0;j<width;j++)
    {
        System.out.print(“*”);
    }
    System.out.println();
}
```

Variable Values

length = 2
width = 3;
i = DOES NOT EXIST
j = DOES NOT EXIST

Console

File Edit View Help

Closer Look

```
for(int i=0;i<length;i++)
{
    for(int j=0;j<width;j++)
    {
        System.out.print(“*”);
    }
    System.out.println();
}
```

Variable Values

length = 2
width = 3;
i = DOES NOT EXIST
j = DOES NOT EXIST

Console

File Edit View Help

Closer Look

```
for(int i=0;i<length;i++)
{
    for(int j=0;j<width;j++)
    {
        System.out.print("*");
    }
    System.out.println();
}
```

Variable Values

length = 2
width = 3;
i = 0
j = DOES NOT EXIST

Console

Closer Look

```
for(int i=0;i<length;i++)
{
    for(int j=0;j<width;j++)
    {
        System.out.print("*");
    }
    System.out.println();
}
```

Variable Values

length = 2
width = 3;
i = 0
j = DOES NOT EXIST

Console

File Edit View Help

Closer Look

```
for(int i=0;i<length;i++)
{
    for(int j=0;j<width;j++)
    {
        System.out.print("*");
    }
    System.out.println();
}
```

Variable Values

length = 2
width = 3;
i = 0
j = DOES NOT EXIST

Console

Closer Look

```
for(int i=0;i<length;i++)
{
    for(int j=0;j<width;j++)
    {
        System.out.print("*");
    }
    System.out.println();
}
```

Variable Values

length = 2
width = 3;
i = 0
j = 0

Console

File Edit View Help

Closer Look

```
for(int i=0;i<length;i++)
{
    for(int j=0;j<width;j++)
    {
        System.out.print("*");
    }
    System.out.println();
}
```

Variable Values

length = 2
width = 3;
i = 0
j = 0

Console

File Edit View Help

Closer Look

```
for(int i=0;i<length;i++)
{
    for(int j=0;j<width;j++)
    {
      System.out.print("*");
    }
    System.out.println();
}
```

Variable Values

length = 2
width = 3;
i = 0
j = 0

Console

```
*
```

File Edit View Help

Closer Look

```
for(int i=0;i<length;i++)
{
    for(int j=0;j<width;j++)
    {
        System.out.print("*");
    }
    System.out.println();
}
```

Variable Values

length = 2
width = 3;
i = 0
j = 1

Console

```
*
```

File Edit View Help

Closer Look

```
for(int i=0;i<length;i++)
{
    for(int j=0;j<width;j++)
    {
        System.out.print("*");
    }
    System.out.println();
}
```

Variable Values

length = 2
width = 3;
i = 0
j = 1

Console

*

File Edit View Help

Closer Look

```
for(int i=0;i<length;i++)
{
    for(int j=0;j<width;j++)
    {
        System.out.print("*");
    }
    System.out.println();
}
```

Variable Values

length = 2
width = 3;
i = 0
j = 1

Console

**

Closer Look

```
for(int i=0;i<length;i++)
{
    for(int j=0;j<width;j++)
    {
        System.out.print("*");
    }
    System.out.println();
}
```

Variable Values

length = 2
width = 3;
i = 0
j = 2

Console

```
**
```

File Edit View Help

Closer Look

```
for(int i=0;i<length;i++)
{
    for(int j=0;j<width;j++)
    {
        System.out.print("*");
    }
    System.out.println();
}
```

Variable Values

length = 2
width = 3;
i = 0
j = 2

Console

**

File Edit View Help

Closer Look

```
for(int i=0;i<length;i++)
{
    for(int j=0;j<width;j++)
    {
        System.out.print("*");
    }
    System.out.println();
}
```

Variable Values

length = 2
width = 3;
i = 0
j = 2

Console

```
***
```

File Edit View Help

Closer Look

```
for(int i=0;i<length;i++)
{
    for(int j=0;j<width;j++)
    {
        System.out.print("*");
    }
    System.out.println();
}
```

Variable Values

length = 2
width = 3;
i = 0
j = 3

Console

```
***
```

Closer Look

```
for(int i=0;i<length;i++)
{
    for(int j=0;j<width;j++)
    {
        System.out.print("*");
    }
    System.out.println();
}
```

Variable Values

length = 2
width = 3;
i = 0
j = 3

Console

```
***
```

File Edit View Help

Closer Look

```
for(int i=0;i<length;i++)
{
    for(int j=0;j<width;j++)
    {
        System.out.print("*");
    }
    System.out.println();
}
```

Variable Values

length = 2
width = 3;
i = 0
j = DOES NOT EXIST

Console

***

File Edit View Help

Closer Look

```
for(int i=0;i<length;i++)
{
    for(int j=0;j<width;j++)
    {
        System.out.print("*");
    }
    System.out.println();
}
```

Variable Values

length = 2
width = 3;
i = 1
j = DOES NOT EXIST

Console

```
***
```

File Edit View Help

Closer Look

```
for(int i=0;i<length;i++)
{
    for(int j=0;j<width;j++)
    {
        System.out.print("*");
    }
    System.out.println();
}
```

Variable Values

length = 2
width = 3;
i = 1
j = 0

Console

```
***
```

File Edit View Help

Closer Look

```
for(int i=0;i<length;i++)
{
    for(int j=0;j<width;j++)
    {
        System.out.print("*");
    }
    System.out.println();
}
```

Variable Values

length = 2
width = 3;
i = 1
j = 0

Console

```
***
```

File Edit View Help

Closer Look

```
for(int i=0;i<length;i++)
{
    for(int j=0;j<width;j++)
    {
        System.out.print("*");
    }
    System.out.println();
}
```

Variable Values

length = 2
width = 3;
i = 1
j = 0

Console

```
***
*
```

File Edit View Help

A Few Steps Later

File Edit View Help

Closer Look

```
for(int i=0;i<length;i++)
{
    for(int j=0;j<width;j++)
    {
        System.out.print(“*”);
    }
    System.out.println();
}
```

Variable Values

length = 2
width = 3;
i = DOES NOT EXIST
j = DOES NOT EXIST

Console

```
***
***
```

Loop Usages

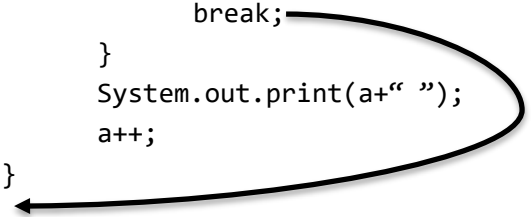
- While
  - Body runs 0 to many times
  - Great for “ask-before-iterating”
- Do-While
  - Body runs 1 to many times
  - Great for “ask-before-iterating”
- For-Loop
  - Body runs a countable number of times
  - Great for “count-controlled” situations

Break Statement

- The statement “break” immediately stops a loop.
- Once the break statement is reached the it will run the next statements outside the body of the loop.
  - “Jumps out of the loop”
- For nested loops it stops the loop whose body the break statement is found.

Break Example

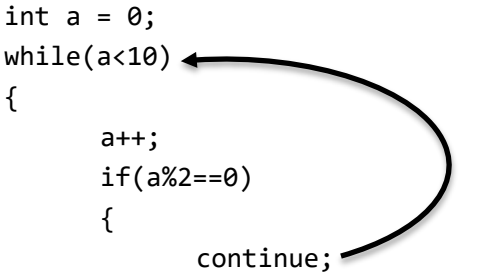
```
int a = 0;
while(true)
{
	if(a >= 5)
	{
		break;
	}
	System.out.print(a+“ ”);
	a++;
}

//Output: 0 1 2 3 4
```



Continue Statement

- The statement “continue” stops the current loop iterations and starts a new one.
- Once the continue statement is reached it immediately starts the loop again.
  - “Jumps back to the start of the loop and continues”
- For nested loops it continues the loop whose body the continue statement is found.

Continue Example



```
int a = 0;
while(a<10)
{
        a++;
        if(a%2==0)
        {
                continue;
        }
        System.out.print(a+ “ ”);
}
//Output: 1 3 5 7 9
```

Sentinel Values

- Special value(s) used signal the end of an algorithm.
- We can use these to stop loops.
- Sentinel values should be selected so that they are distinct from other valid values.

Sentinel Value Example

```
//Find the average of positive values
int value = 0;
int sum = 0;
int count = 0;
while(true)
{
	value = keyboard.nextInt();
	if(value < 0)//The sentinel values are negative
	{
		break;
	}
	sum += value;
	count++;
}
int average = sum / count;
System.out.println(average);
//If the input was 2 4 6 8 10 -1 then it would output 6
```