JAVA REPORT

Date:	**9/06/2020**	**Name:**	**SAFIYA BANU**
Course:	**JAVA**	**USN:**	**4AL16EC061**
Topic:	1. Programming core java 1. A Hello World Program 2. Using Variables 3. Strings: Working With Text 4. While Loops 5. For Loops 6. "If" 7. Getting User Input 8. Do ... While 9. Switch 10. Arrays	**Semester & Section:**	**8TH B**
Github Repository:	**Safiya-Courses**		

FORENOON SESSION DETAILS

PROGRAMMING WITH CORE JAVA

A "Hello, World!" is a simple program that outputs Hello, World! on the screen. Since it's a very simple program, it's often used to introduce a new programming language to a newbie.

Variables

Variables are containers for storing data values.

In Java, there are different types of variables, for example:

- **String - stores text, such as "Hello". String values are surrounded by double quotes**

- **int - stores integers (whole numbers), without decimals, such as 123 or -123**
- **float - stores floating point numbers, with decimals, such as 19.99 or -19.99**
- **char - stores single characters, such as 'a' or 'B'. Char values are surrounded by single quotes**
- **boolean - stores values with two states: true or false**

SYNTAX

type variable = value;

Strings

Strings are used for storing text.

A String variable contains a collection of characters surrounded by double quotes:

EXAMPLE:

```
String greeting = "Hello";
```

Loops

Loops can execute a block of code as long as a specified condition is reached.

Loops are handy because they save time, reduce errors, and they make code more readable.

While Loop

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

Syntax

```
while (condition) {
  // code block to be executed
}
```

In the example below, the code in the loop will run, over and over again, as long as a variable (i) is less than 5:

Example

```
int i = 0;

while (i < 5) {

  System.out.println(i);

 i++:

}
```

For Loop

When you know exactly how many times you want to loop through a block of code, use the for loop instead of a while loop:

Syntax

```
for (statement 1; statement 2; statement 3) {

 // code block to be executed

}
```

Statement 1 is executed (one time) before the execution of the code block.

Statement 2 defines the condition for executing the code block.

Statement 3 is executed (every time) after the code block has been executed.

The example below will print the numbers 0 to 4:

Example

```
for (int i = 0; i < 5; i++) {

  System.out.println(i);

}
```

Conditions and If Statements

Java supports the usual logical conditions from mathematics:

- Less than: `a < b`
- Less than or equal to: `a <= b`
- Greater than: `a > b`
- Greater than or equal to: `a >= b`
- Equal to `a == b`
- Not Equal to: `a != b`

if Statement

Use the `if` statement to specify a block of Java code to be executed if a condition is `true`.

Syntax

```
if (condition) {
  // block of code to be executed if the condition is true }
```

EXAMPLE `int x`

```
= 20; int y =
```

```
18; if (x > y)

{


  System.out.println("x is greater than y");

}
```

Switch Statements

Use the `switch` statement to select one of many code blocks to be executed.

Syntax

```
switch(expression) {

case x:


    // code block

break;   case y:


    // code block
break;
  default:

   // code block

}
```

While Loop

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

Syntax

```
while (condition) {

  // code block to be executed

}
```

In the example below, the code in the loop will run, over and over again, as long as a variable (i) is less than 5:

Example

```
int i = 0; while

(i < 5) {


  System.out.println(i);

i++; }
```

Arrays

Arrays are used to store multiple values in a single variable, instead of declaring separate variables for each value.

To declare an array, define the variable type with **square brackets**:

```
String[] cars;
```