Section 5: Control Flow Statements

Topics Covered:

Switch Statement

For Statement

While Statement

Do-while Statement

Continue and break

Parsing Values from a string

Switch Statement

The same as an if else

Switch is better when you are testing different values for the same variables

Example 1

```
int switchValue = 1;

switch(switchValue){
   case 1:
      System.out.println("Value was 1");
      break;
   case 2:
      System.out.println("Value was 2");
      break;
   case 3: case 4: case 5:
      System.out.println("was a 3 or a 4 or a 5");
      System.out.println("Actually it was a " + switchValue);
      break;
   default:
      System.out.println("was not 1 or 2");
      break;
}
```

Example 2:

```
String month = "January";
```

```
switch(month.toLowerCase()){
   case "january":
      System.out.println("January");
      break;
   case "february":
      System.out.println("February");
      break;
   default:
      System.out.println("Not sure");
      break;
}
```

<u>For Statement</u>

```
For (initialization; condition; increment/ decrement){

}
```

<u>Example:</u>

```
for(int i =0; i<5; i++){
   System.out.println("Loops " + i);
}
```

<u>While Statement</u>

```
While(condition){

// increment or decrement

}
```

<u>Example:</u>

```
int count =0;
while(count !=6){
   System.out.println("Count value is " + count);
   count++;
}
```

Do-while Statement

It will execute at least once

Example:

```
int count =1;
do{
   System.out.println("count value was " + count);
   count++;
}while(count <= 6);
```

Continue and break

As in C# and C++

We can interrupt the loop using continue and break statements

Continue: The loop will bypass the part of the code block that is below the continue keyword and go to the next iteration

Break: will exit the loop

Parsing Values from a string

Converting a string to a different data type and vice versal

```
String numAsString = "2018";
```

```
int num = Integer.parseInt(numAsString);
double numD = Double.parseDouble(numAsString);
```

Reading User Input

Scanner allows us to read user inputs

Nb: after you read a number with a scanner there must be a next line method call to handle the enter key

```
Scanner scanner = new Scanner(System.in);


System.out.println("Enter your year of birth: ");
// takes an input and puts true if the input is int and false otherwise
// The hasNextInt() method returns true if and only if this scanner's next token is a valid int value
Boolean hasNextInt = scanner.hasNextInt();

//scanner has a method that will parse the string to int for us
int yearOfBirth = scanner.nextInt();
scanner.nextLine(); // handle next line character (enter key)

String name = scanner.nextLine();

scanner.close();
```

MORE: [Java User Input (Scanner class) (w3schools.com)]

The `Scanner` class is used to get user input, and it is found in the `java.util` package.

To use the `Scanner` class, create an object of the class and use any of the available methods found in the `Scanner` class documentation. In our example, we will use the `nextLine()` method, which is used to read Strings:

Example

```
import java.util.Scanner;  // Import the Scanner class


class Main {
  public static void main(String[] args) {
    Scanner myObj = new Scanner(System.in);  // Create a Scanner object
    System.out.println("Enter username");
    String userName = myObj.nextLine();  // Read user input
    System.out.println("Username is: " + userName);  // Output user input
  }
}
```

Input Types

In the example above, we used the `nextLine()` method, which is used to read Strings. To read other types, look at the table below:

Method	Description
`nextBoolean()`	Reads a `boolean` value from the user
`nextByte()`	Reads a `byte` value from the user
`nextDouble()`	Reads a `double` value from the user
`nextFloat()`	Reads a `float` value from the user
`nextInt()`	Reads a `int` value from the user
`nextLine()`	Reads a `String` value from the user

`nextLong()`	Reads a `long` value from the user
`nextShort()`	Reads a `short` value from the user

In the example below, we use different methods to read data of various types: