

# CSCE 156 – Lab: Conditionals & Repetition

---

## *Java Version*

### 0. Prior to the Laboratory

1. Review the laboratory handout
2. Read if-then-else tutorial:  
<http://download.oracle.com/javase/tutorial/java/nutsandbolts/if.html>
3. Read switch/case tutorial:  
<http://download.oracle.com/javase/tutorial/java/nutsandbolts/switch.html>
4. Read for loop tutorial:  
<http://download.oracle.com/javase/tutorial/java/nutsandbolts/for.html>
5. Read while/do while loop tutorial:  
<http://download.oracle.com/javase/tutorial/java/nutsandbolts/while.html>

### 1. Lab Objectives

Following the lab, you should be able to:

- Use if-then-else statements to control the logical flow of the program.
- Use switch-case statement to control the logical flow of the program.
- Use for/while loops to implement repetition statements in your program.
- Write complex Java programs that require conditional logical statements and or loops.

### 2. Lab Topics

- if-then-else/switch statements
- for/while/do while loops

### 3. Problem Statement

Java provides standard control structures for conditionals and repetition. Specifically, Java provides the usual if-then-else statements and while, for, and do-while loops. The syntax for these control structures should look familiar; some examples:

```
if(condition1) {  
    //DO SOMETHING  
} else if(condition2) {  
    //DO SOMETHING ELSE  
} else {
```

## Lab Handout: Conditionals & Repetition

```
//OTHERWISE
}

for(int i=0; i<n; i++) {
    //DO SOMETHING
}

int i=0;
while(i<n) {
    //DO SOMETHING
    i++;
}

int i=0;
do{
    //DO SOMETHING
    i++;
} while(i<n);
```

In addition, Java provides a foreach-loop, also referred to as an *enhanced for-loop*, for iterating over collections (classes that implement the `Iterable` interface) or elements in an array. This feature is mostly for convenience. The following examples illustrate this loop's usage.

```
String arr[] = new String[10];
...
for(String s : arr) {
    System.out.println(s);
}
```

### Activity 1: Sum of Natural Numbers

Natural numbers are the usual counting numbers; 1, 2, 3, ... In this exercise you will write several loops to compute the sum of natural numbers 1 thru  $n$  where  $n$  is read from the command line. You will also write a foreach loop to iterate over an array and process data.

#### Instructions

1. Download the `Natural.java` file from Blackboard and import it into a project in Eclipse.
2. You'll note that code to read in  $n$  has already been provided for you. An array mapping integer values 0 thru 10 to text values has also been created for you.
3. Write a for-loop and a while-loop to compute the sum of natural numbers 1 thru  $n$  and output the answer.
4. Write an enhanced for-loop loop to iterate over the elements of the `zeroToTen` array. As you iterate over the elements, concatenate each string, delimited by a single space to a result string and print the result at the end of the loop. Your result should look something like the following:  
zero one two three four five six seven eight nine ten
5. Demonstrate your working code to a lab instructor and have them sign your worksheet.

## Activity 2: Child Tax Credit

When filing for federal taxes, a credit is given to tax payers with dependent children according to the following rules. The first dependent child under 18 is worth a \$1000.00 credit. Each dependent child younger than 18 after the first is worth a \$500 tax credit each. You will complete a Java program to output a table of dependent children, how much each contributes to a tax credit, and a total child tax credit. Your table should look *something* like the following.

| Child         | Amount    |
|---------------|-----------|
| Tommy (14)    | \$1000.00 |
| Richard (12)  | \$500.00  |
| Harold (21)   | \$0.00    |
| Total Credit: | \$1500.00 |

### Instructions

1. Download the `Child.java` and `ChildCredit.java` files from Blackboard to a project in Eclipse.
2. The `Child` class has already been implemented for you. Note how the `Child` class is used in the `ChildCredit` main method; several instances of children have been created and placed into an array.
3. Write code to iterate over the array, compute the child tax credits and output a table similar to the one above. Note: to call a method on an instance of the `Child` class, use the following syntax: `kid.getAge()`
4. Answer the questions in your worksheet and demonstrate your working code to a lab instructor.

### Advanced Activity (Optional)

Use the `String.format` method to reformat the output of the Child Tax Credit program to print every piece of data in its own column.