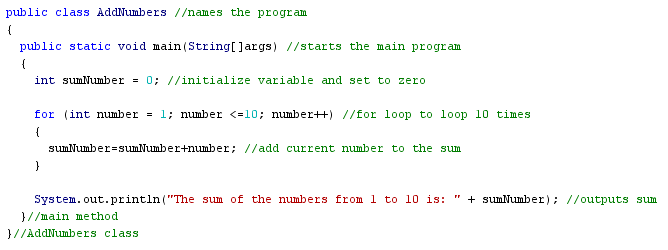
The main power of a computer resides in its ability to do a number of operations very quickly and accurately. A computer may be required to repeat these operations, over and over without becoming bored or inaccurate, until the required work is complete. The repeated occurrence of an operation or event in a computer program is known as repetition or, in programmer jargon, "looping". Repetition is one of the three basic programming structures and is an essential part to any useful computer program.

Like all programming languages, Java has a number of constructs that are used for repetition. The two basic types of loops that we will learn are the ***for*** loop and the ***while*** loop. To compare the use of the two repetition structures we will solve the same problem (For Loops in Part 1, and While Loops in Part 2)

**Task 1: (from Powerpoint)**

1. **Explain the difference between a ‘while loop’ and a ‘for loop’, and when you would use each.**
2. **What 3 things does a for loop need?**
3. **When should you use a for loop in your code?**
4. **Make notes as needed on the syntax of for loops**

**Example of For Loop:**  
**Some notes:**

* The initialization statement must have the type of variable (usually an integer and referred to as the "*index*"), as well as its first value (in this case,**int**number = 1 ). The ***for*** statement automatically declares the variable (number). The *index* variable may also be referred to as a control variable or counter.
* The condition statement determines whether or not the ***for*** loop is executed, and must have some type of comparison (<, >, =,etc.) as part of it.
* The increment/decrement statement in this example could have been rewritten as number=number +1, but it is easier to use number++ as a short form. (number-- would decrease the variable by one each time)

**Task 2: Modifying AddNumbers**

1. **Type in the program AddNumbers above, compile, correct any errors and run.**
2. **Change the program to add all the numbers from 1 to 20.**
3. **Change the program to add all the EVEN numbers from 1 to 20.**

**Task 3: Writing new programs**

1. **Write a program that asks the user their name. Output their name 5 times**
2. **Modify your program from (a) to ask the user their name and their favourite number. Output their name as many times as their favourite number**
3. **Modify your program from (b) to check that their favourite number is an appropriate value (i.e. not negative, not larger than a given value (100?)). If it is outside your constraints, give an error message.**
4. **Write a program called TimesTables to output a set of times tables from 1 to 10 for any number. Sample output shown below – accept user input for the times table that they want displayed and use that input for calculations.**

Sample output:  
> 3x1=3  
> 3x2=6  
>3x3=9  
{etc… showing up to 3x10=30}

1. **Write a program to output a list of symbols related to the user’s integer input**

Sample output (if user entered “7”)

>\*\*\*\*\*\*\*

(note for this you may wish to use the System.out.print command vs println)

1. **Get 5 integers from 1 to 20 from the user. Use them to print out a bar graph**

Sample output (if user entered “7, 3, 2, 5, 9”)

>\*\*\*\*\*\*\*  
>\*\*\*  
>\*\*  
>\*\*\*\*\*  
>\*\*\*\*\*\*\*\*\*

1. **3U ONLY - modify your code from (f) to allow the user to choose a number of “bars” from 2-10. Also add a check to ensure that they are entering appropriate values.  
   You may need to do a “nested” loop - a loop within a loop.**
2. **(3U ONLY) The Fibonacci sequence is found by adding the two previous terms:  
   1, 1, 2, 3, 5, 8, 13, 21, …  
   Write a program to ask the user for how many terms they want, and print out that many terms of the Fibonacci sequence.**