BRAC University
Department of Computer Science and Engineering
CSE110 (Programming Language-I)
Lab 5

Objective

So far in this class, we have learnt about the basic building blocks of all java programs - **literals, data types, variables and operators**. For the first two weeks of labs, we focused solely on **programs that have a completely sequential structure**. These programs followed all steps one after another from start to end, without skipping or repeating any step. In lab 3, we saw one way of writing programs with a little more flexible control flow – the conditional statement. Using conditional statements, we could make our programs take one of multiple possible paths, based on the condition of the program at the time of this statement.

Today, we will first write a few more programs that are either completely sequential, or use some combination of conditional statements. Then, we will introduce you to programs that allow us to have a set of steps repeated a certain number of times. This is referred to as a **loop or iteration**. As you will from some of the programs you write today, being able to repeat a certain number of steps allows us to write programs that are much more expressive and effective.

Exercise Set A

Task 1:
Write a java program that reads five numbers as input from the user, and prints whether the numbers are odd or even.

Task 2:
Write a java program that reads five numbers from the user, and prints their average.

Task 3:
Write a java program that prints the first ten positive integers.

Task 4:
Write a java program that prints the first ten even positive integers.

Hopefully, none of the tasks above were too difficult. For all of them, the solution method is quite simple and does not need anything more advanced that what we have already covered. In fact some of them are exactly identical to programs you have written in the previous labs. However, consider what happens if we change the scale of the problem a little. What happens if task 1 and task 2 ask you to deal with 20 numbers instead of just 5? What about if the range for tasks 3 and 4 is increased to 100 instead of 10?

The nature of the problems has not changed. The basic solution procedure has not changed either. The rule for computing the average of five numbers is no different from that for computing twenty numbers.

What has changed is the scale of the problem. In order to solve the modified problems, we have to repeat a set of basic steps a certain number of times, as specified by the problem statement. Imagine this – to solve task 1 or 2, we have to rewrite the input, calculation and output steps twenty times! Such repetition is boring and error-prone. We need some convenient way to have a set of statements repeat a certain of number of specified times. This is where **loops** come in.

Exercise Set B

Task 5:
Repeat task 1 for twenty numbers.

Task 6:
Repeat task 2 for twenty numbers.

Task 7:
Repeat task 3 for the first hundred positive integers.

Task 8:
Repeat task 3 for the first hundred even positive integers.

Task 9:
Write a java program that prints the following sequences of values using loops:

a) 24, 18, 12, 6, 0, -6
b) -10, -5, 0, 5, 10, 15, 20
c) 18, 27, 36, 45, 54, 63
d) 2, -4, 6, -8, 10, -12

Task 10:
Write a Java program which adds all numbers that are multiples of both 7 and 9 up to 600.

Task 11:
Write a Java program which adds all numbers that are multiples of either 7 or 9 up to 600. Ensure that numbers like 63 are added only once in the sum.

Task 12:
Write a Java program which adds all numbers that are multiples of either 7 or 9 but not both, up to 600.