North South University

Department of Electrical and Computer Engineering

CSE 215L (Programming Language II Lab)

Lab 4: Methods, Method Overloading

Objective:

- **Method with Conditional Statements:** Learn how to incorporate conditional statements (if, else if, else) within methods to create flexible and responsive functions based on specified conditions.
- **Loop Structures within Methods:** Explore the integration of loop structures (for, while, do-while) within methods to perform iterative tasks efficiently and maintain modular code.
- **Method Overloading:** Understand the concept of method overloading, allowing the creation of multiple methods with the same name but different parameter lists. Explore its application for versatility in method usage.
- **Array Manipulation in Methods:** Learn to pass arrays as parameters to methods and manipulate array elements within the method for tasks like sorting, searching, or filtering.

Methods:

First, a method is a code block or a portion executed upon being called. Methods take parameters to be used inside it for various purposes, or they can take no parameters and just run.
Methods can either return a value or not, depending on the return type. Remember that we have used the keyword void a lot. If a method is written as void <method>, it does not return anything, rather, it just plain runs.
How a method is made is given below:

```
public static int max(int num1, int num2){
	int result;
	if (num1 > num2){
		result = num1;
	} else {
		result = num2;
	}
	return result
}
```

Here, the method is named ***max*** and takes in two parameters that are used inside. Notice that we have written ***public static int*** here, why, let me explain:
public -> can be accessed from anywhere
int -> returns an ***integer*** value
static -> so that the method represents the whole class instead of a class object

Inside a method, you can do whatever you need to do, for example, the previous method used conditional statements but you can also use loops in a method if you feel that writing the same thing more than once in the main method is not worth anyone's time.

Method Overloading:

Can you make two or more separate methods with the same name? It is possible due to method overloading to make methods of the same name but with a different set of parameters. Now, look at the following two methods given below:

int getResult(int a, int b)	double getResult(double a, double b)
`public static int getResult(int a, int b){ return a+b; }`	`public static double getResult(double a, double b){ return a-b; }`

Now, as you see, both of these methods have the same name but different sets of parameters, the first having two integer parameters and the other having two double parameters. However, depending on the type or/and number of parameters, the implementation of the methods could be different. Thus, methods could be overloaded depending on the situation.

Practice Problems:

1. Create a method named int sumOfSquares(int n) which returns the summation of squares from 1 to n where the value of n is from the user input.
2. Recreate the max(int m, int n) method from the Math class (covered in Lab 3), but this time, YOU ARE BANNED FROM USING MATH CLASS FOR THIS ONE.
3. Declare an integer array of size 10, then assign the values as you see fit. Then, you find the summation of these numbers you had just entered, followed by finding the maximum of those numbers.
4. Recreate the pattern printing exercise (Lab 4 Homework Question 1), but this time, use the method void createPattern(char a, char b), where a and b could be any character of your choosing.
5. Print the Fibonacci Sequence from 0 to n using a method where n is user input. The method created should be named void Fibonacci(int n) which returns no value, just prints. Fibonacci series progress in this manner -> 0, 1, 1, 2, 3, 5, 8, 13, 21…..
6. Declare a double array of size 10, then assign the values as you see fit. After that, if the numbers are NOT IN proper order, sort them in ascending order (small to large). For example, if an array has a value of 5 6 2 8 1, the sorted array will be 1 2 5 6 8.
7. This time, create THREE methods named double priceCalculation(), where the parameter sets and implementation is given as follows:
    a. The first method takes in the price of one product as a double parameter and adds half of the price, which is then returned
    b. The second method takes in the price of two products and adds the prices together, which is then returned.

c. The third method takes in the price of three products and does the following things:

   I. If the total price is below 200, increase the total price by 50 and return the newly updated total price.

   II. If the total price is between 200 and 400, just return the total price.

   III. If the total price is above 400, then lower the total price by 50 and return the newly updated total price.

Print the returned values from ALL THREE METHODS to check whether you feel you are on the right track.