**ITCS 1212L**

**Pre-Lab 1**

**From Source Code to Executable and Algorithms**

**Learning Objectives:**

* To understand the basics of program design and algorithm development
* To learn the basics of an editor and compiler
* To learn the process of converting a source code to executable code
* To learn the difference between syntax errors and logic errors
* Practice developing algorithm

**Answer these questions by filling the blanks:**

1. Compilers detect \_\_\_\_\_\_\_\_\_\_\_\_\_\_ errors.
2. Usually the most difficult errors to correct are the \_\_\_\_\_\_\_\_\_\_\_\_\_ errors, since they are not detected in the compilation process.
3. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ code is the machine code consisting of ones and zeroes that is read by the computer.
4. If we compare programming to cooking the recipe is like \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.
5. What are the main components of a typical computer system?
6. What is machine language and how is that related to these components?
7. What is natural language? Give examples of it.
8. Compare how our brain processes natural language instructions with how processor processes the machine language.
9. What is high-level programming language (HLL)? At least name three different HLL programming languages.
10. What is the programming language that we use in this class? List few advantages of this language.
11. What are the main components of CPU?
    1. What is the language of CPU?
    2. What are the main tasks of CPU?
12. What are the two basic processing capabilities of processors? Give examples of each category.
13. What is ‘source code’? Give an example of a source code.
14. How do you create the source code? Give at least name of 3 different tools to enter source code with them.
15. What is ‘the executable code’?
    1. How do you generate the executable code?
    2. Name at least 2 different tools that you can generate executable codes from source code with them for the language of this class.
16. What does IDE stand for? Name at least 2 different IDEs that we will use in this course.
17. cout << “Hello World!” << endl; is an example of \_\_\_\_\_\_\_\_\_\_\_\_\_\_ fundamental instruction.

**Written Exercise:**

1. Let’s develop an algorithm:

**An algorithm is a set of clear, concise, correct, step-by-step sequence of steps to solve a problem.** Let’s look at an algorithm for washing your hands:

1. Turn cold water tap clockwise twice.
2. Place one hand under soap dispenser.
3. Pump the top of the soap dispenser with the other hand several times.
4. Place both hands under tap.
5. Rub together till soap clears from hand.
6. Remove hands from under the tap.
7. Turn cold water tap counter-clockwise until it does not turn anymore.

Writing an algorithm for processing data is very similar to above steps. For example, let’s assume we want to multiply two numbers also decrement the first one and subtract it from the first number. You can develop the following list of steps for above actions.

* + - 1. Declare two numbers as integers, say x and y.
      2. Next, multiply them and assign the result to another integer z.
      3. Decrement first input number.
      4. Update the second number y by subtracting x.