

Please read 1.3–1.6, Figure 1.6, 3.1–3.5 of the textbook and then answer the following, trying not to look at your notes or at the textbook. Quiz #1, on Thursday 18th January, will consist of questions taken or inspired from the Part I of this homework and from the lab. Project #1, presented in Part II, 2, is due Thursday 25th January, before 11:30 PM.

## Part I — Questions

1. List five softwares, and three hardware components of a computer.
2. List four programming languages.
3. What is a GUI? What is a CLI?
4. What is the “input” of a program? What is its “output”?
5. What, if any, is the difference between a compiler and an assembler?
6. A machine code (or machine language program) is made of binary numbers, circuits, classes, or compilers?
7. Using the analogy developed in class, if we say that an instance is a cookie, then the class is
  1. the cook
  2. the floor
  3. the cookie-cutter
8. An object contains data and performs procedures, but is “method” the data, and “attributes” the procedure, or is it the other way around?
9. What is encapsulation, and what is data hiding?
10. Briefly describe what inheritance is.
11. What is the relationship between an object and a class? Does two objects have the same data? The same underlying structure?
12. What happen when the source code you’re giving to the compiler has syntax error? Is the compiler case-sensitive?
13. Suppose I replace every white space in your source code with two white spaces. Will you program still compile? Why, or why not?
14. Which of the followings, if any, are keywords?  
`Welcome1          public          91.3          apples          int          "I’m a string"`
15. Which of the following are programmer-defined names (or identifiers)?  
`BankAccount    class    =    91.3    apples    int    itemPerCapita    statement`
16. Why are variables called “variables”?
17. What is string interpolation?
18. What is the difference between 3, and "3"?
19. In Visual Studio, what is a project? What is a solution?

20. In a C# program, comments start with `\\` (double backslash) or with `//` (double (forward) slash)? Do they have to end with a `;` (semicolon)?
21. What is the limitation, if any, to the number of methods you can have per classes? Why is the method called `Main` special?
22. What is a namespace?
23. What is the difference, if any, between the `WriteLine` and `Write` methods?
24. Which of the following are correct identifier names?  
`$myHome3` `class` `my%variable` `91.3` `ANewHope` `9_train` `_ThisIsAVariable` `statement`  
Is the name `myVariable` the same as `myvariable`? If not, why?
25. Which one(s) of the following, if any, is a correct assignment (assuming that `variable`, `x` and `apples` have been declared as `int` variables)?  
`5 => variable;`  
`x=5;`  
`apples= 23`  
`x <= 23;`  
`variable =1,890;`
26. Does a variable have attributes and methods? If your answer is yes, give one example of each.

## Part II — Programming Exercises

1. As a warm-up, look at the following code. There are 4 errors that will prevent it from compiling. Can you spot them all?

```
// My first attempt.
using System

class Wel
{
    static void Main();
{
    Console.WriteLine("Welcome \n to the lab!");
}
```

2. In Visual Studio, create a project named “Proj01”. Write your name and the date in a delimited comment at the very beginning of your code. In the main method, declare a variable and store your augusta username in it. Then, your program should print the content of that variable, followed by “would like to know your favorite number. Please enter it:”. After this, the program should read an integer from the user, multiply it by 2, and print it on the screen.

Once your project is completed, compile without error nor warning, and can be executed, compress it into a zip file named “lname\_fname.zip”, where “lname” (resp. “fname”) is your last name (resp. first name). Upload this zip on D2L before **Jan 25, 2018 11:30 PM**, in the “Proj01” assignment submission folders.

Further details will be given in Lab, but make sure you follow those instructions as closely as possible.

