COMP 1202
Object Oriented Programmings (OOP)
Lab Session- online - Week 02

Mehdi Azad

January 16, 2024

Organization

My contact

- `mehdi.azad@georgebrown.ca`

Course Code

- Github: `https://github.com/mabbasiazad/COMP1202`

AtKlass

- code for this session: MWXF
- self-register: 0G9I (zero)

New code editor

I'd like to shift from *Visual Studio 2022* to *Visual Studio Code* (commonly referred as VS Code)

Why?

- work well both on mac and windows
- it's a very light editor and easy to install
- it's not just for C# you can code in Java, Python, Scala, etc.

How to install?

- google **dotnet** and download .NET
- download visual studio code (blue icon)
- open VS Code and install C# Dev Kit from Extensions

Creating new project and solution for C# in VS Code

go to the folder you want to create your solution on command line and type

```
$ dotnet new sln
```

use **Solution Explorer** to create new projects

Review of week01

```
namespace ExampleConsoleApp
{
    // single line comment
    internal class Program
    {
        /* Main method is the entry point
        of the program (multiple line comment)*/
        static void Main(string[] args)
        {
            Console.WriteLine("Hello, World!");
        }
    }
  }
```

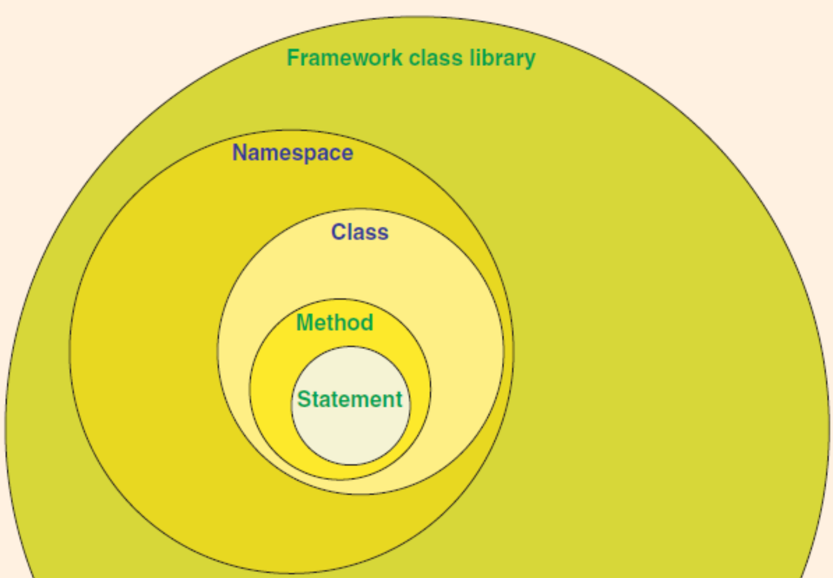
Elements of the C# program

- The same name in different namespaces can have different meanings
- you can remove the curly braces after the namespace and just put semicolon.

```
namespace ExampleConsoleApp;
```

- Typically the application takes the form of a *class* that has a Main method.
- The Main method has to follow a particular convention. it takes single argument (args) of type array of string and it returns nothing (void)
- Inside the Main method I can write my code (statements)

Relationship among C# elements



Useful methods

Console.WriteLine("msg")	writes the string msg to the monitor Then, advances to the **next line**
Console.Write("msg")	Same as WriteLine(); but not going to the **next line**
Console.ReadLine()	accepts **multiple** characters until press **Enter** key returns **string** type
Console.ReadKey()	accepts **single** character. keep the output screen displayed until the user presses a key on the keyboard does not require **Enter**
Console.Read()	accepts **single** character and return its **ASCII Code** So data type should be **int**

Exercise 1 (from textbook chapter 2)

Write a program that converts a temperature given in Celsius into Fahrenheit. Test the program by performing a compile-time initialization of 32 for the original Celsius value. Display the values number aligned. The original temperature should show no digits after the decimal. One position following the decimal should be printed for the converted value. Be sure to provide labels for both values. Go into your source code and change the initialization value to 0. Rerun the application. Select additional test values and rerun the application.

Formula : $(C \times 9 / 5) + 32 = F$

Exercise 2 (from textbook chapter 2)

Design an application that converts miles into feet and its equivalent metric kilometer measurement. Declare and initialize miles to 4.5. Show your miles and kilometers formatted with two positions to the right of the decimal. Feet should both be shown with no positions to the right of the decimal with comma separators. Be sure to provide labels for values and number align them. Once you get that portion running, go into your source code and change the initialization value for miles. Rerun the application and make sure that your values are still number aligned

Formula:

1 mile = 1.6 km

1 mile = 5280 feet

Exercise 3 (from textbook chapter 2)

Write a program that prints the number of quarters, dimes, nickels, and pennies that a customer should get back as change. Declare and initialize all memory locations as integers. On output, show the original change amount as a monetary amount, with two positions to the right of the decimal. Run your program once by performing a compile-time initialization using 92 cents for the value to be converted. Go into your source code and change the 92 to 27. Rerun the application. Be sure to desk check your solutions.