COMP 1202
Object Oriented Programmings (OOP)
Lab Session - Week 01

Mehdi Azad

January 12, 2024

Organization (1)

Instructor

Mehdi Azad

Email `mehdi.azad@georgebrown.ca`

Course website

- Syllabus, Assignments, slides, lecture notes, due dates: D2L Brightspace | George Brown College

Textbook

- C# Programming: From Problem Analysis to Program Design, 5th Edition (Visual Studio 2015) by Barbara Doyle

Organization (2)

AtKlass

- code for this session: ****
- self-register: ****

Grading

- Lab Exercises: 7 X 1% each
- Lab Test 1: 10%
- Lab Test 2: 10%

Software Requirement

Software Requirement

Visual Studio 2022

- Desktop and Mobile –> .NET Desktop development (C#)
- Data Storage and processing

We'll create the App in the class

Software Development Process

Programming is a process of **problem solving**

The methodology to solve computer-related problems

- Problem Specification
- Analyze the problem
- Design a solution
- Write and Implement the code
  - We use C# for this purpose in this course

We can ask ChatGPT to solve our problems, but ChatGPT is a program itself written by programmers following the steps above

Programming Paradigms

- Imperative Programming
    - Procedural Programming
    - Object Oriented Programming

  In both paradigms we break down the program in more manageable pieces (modules) of step by step *instructions*.

  The difference is in the level of modularity. Procedural Programming has *(sub)routines* and Object Oriented Programming has *classes* and *objects*.

- Functional Programming

  Functions are mathematical functions. They don't change anything. they just map inputs to outputs.

Creating C# Console App in Microsoft Visual Studio 2022

We'll create the App in the class

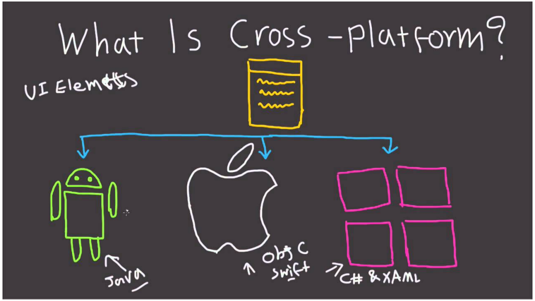
Implement the code (1)

FIRST STEP:

source code are compiled to check for rule violations (list of errors)

the source code is converted into the **Microsoft Intermediate Language**

IL code is between the high level source code and machine language native code



Implement the code (2)

SECOND STEP:

- **just-in-time (JIT) compilation**

- converts the MSIL code into the platform's machine native code.

- This second step is managed by .NET's common language runtime (**CLR**).

- The CLR is included with the **.NET Framework**

- Any computer executing .NET code must have the .NET Framework installed

Write Console App in C#

```
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!");
        }
    }
  }
```

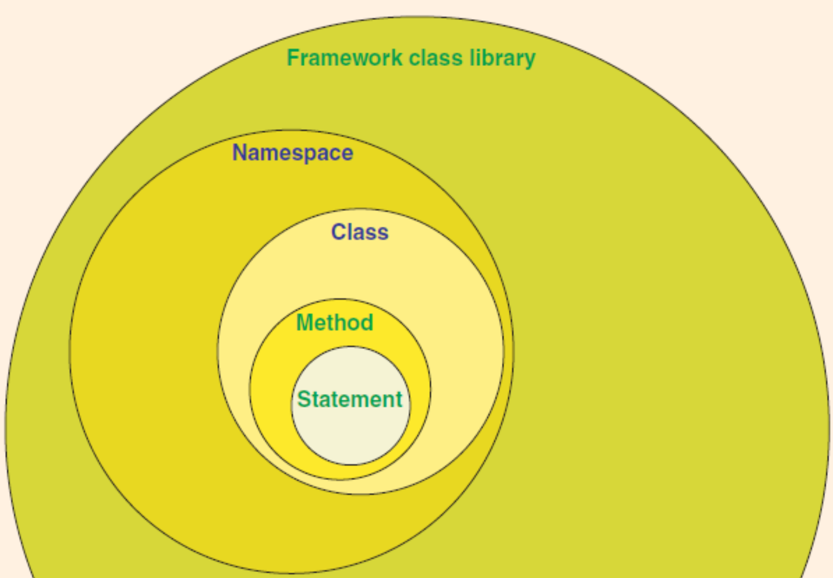
Elements of the C# program

- A *namespace* is also called a context, because 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)

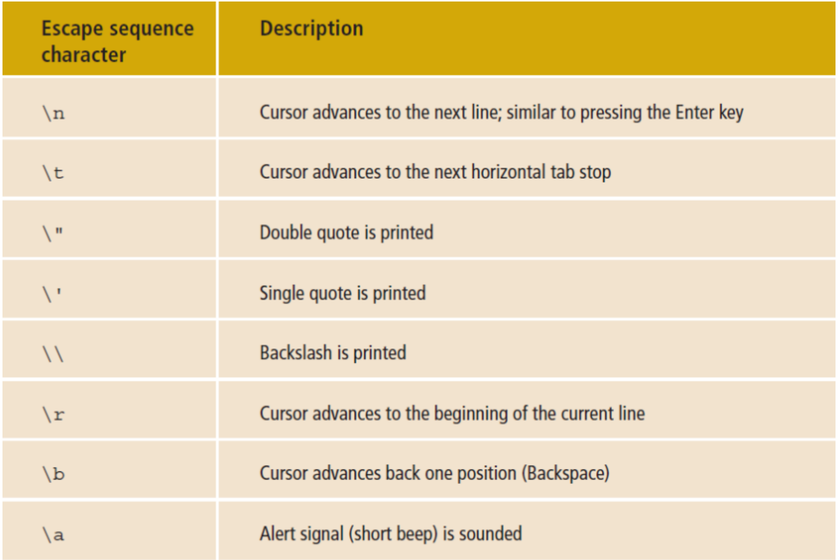
Relationship among C# elements



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
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*

Escape Sequences



Escape sequence character	Description
`\n`	Cursor advances to the next line; similar to pressing the Enter key
`\t`	Cursor advances to the next horizontal tab stop
`\"`	Double quote is printed
`\'`	Single quote is printed
`\\`	Backslash is printed
`\r`	Cursor advances to the beginning of the current line
`\b`	Cursor advances back one position (Backspace)
`\a`	Alert signal (short beep) is sounded

Example

```
static void Main(string[] args)
{
    Console.WriteLine("what is your name?");
    var name = Console.ReadLine();
    var currentdate = DateTime.Now;
    Console.WriteLine($"\nhello, {name}, on
       {currentdate:d} at {currentdate:t}!");
    Console.Write($"\npress any key to exit...");
}
```

note: run on your computer to see the result