

Intro to C++

C#

C++



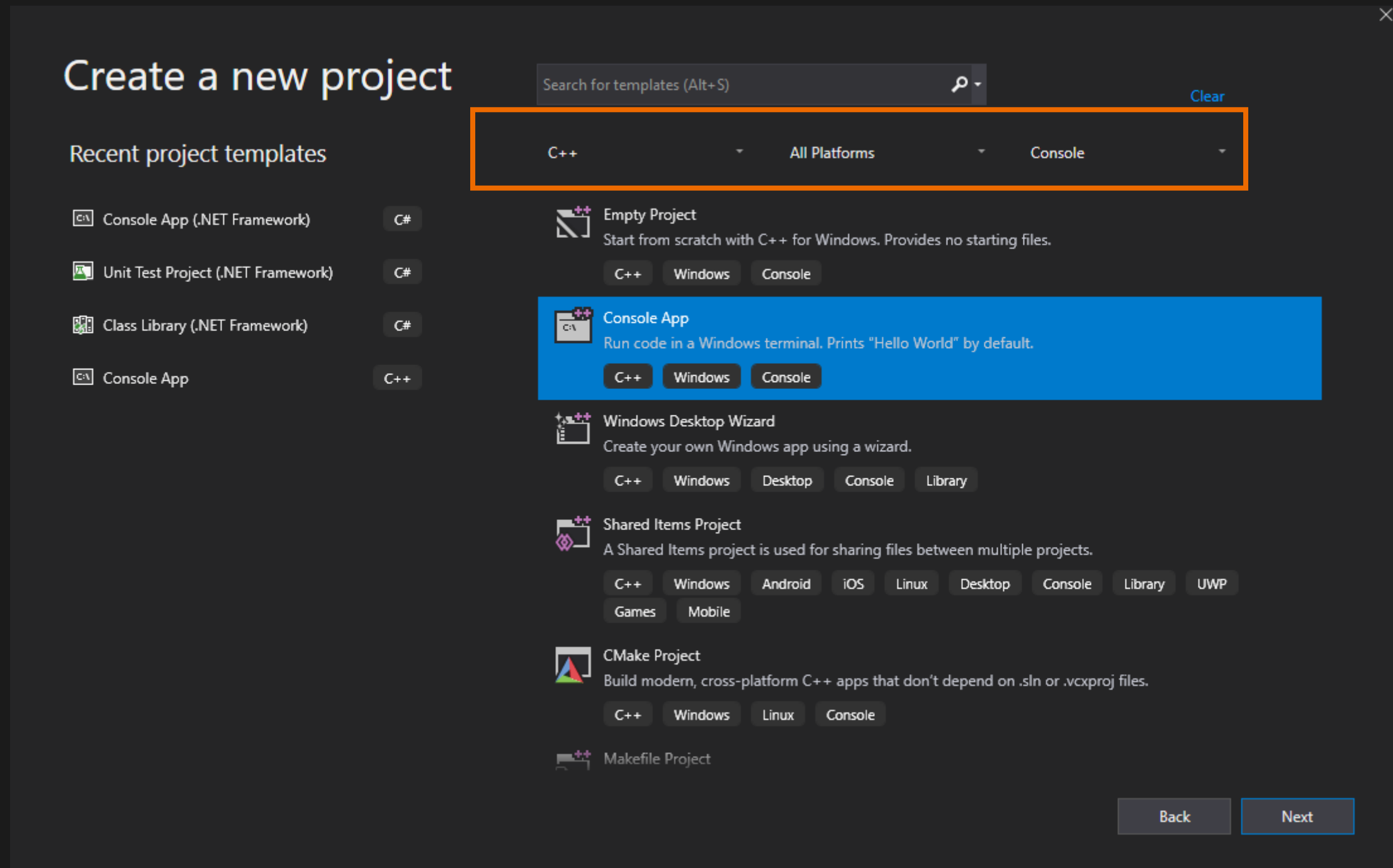
C# vs C++

- C# and C++ are **both C-based** languages
- Therefore, they are **very similar**.
- You **already** know how to code the basics in C++.

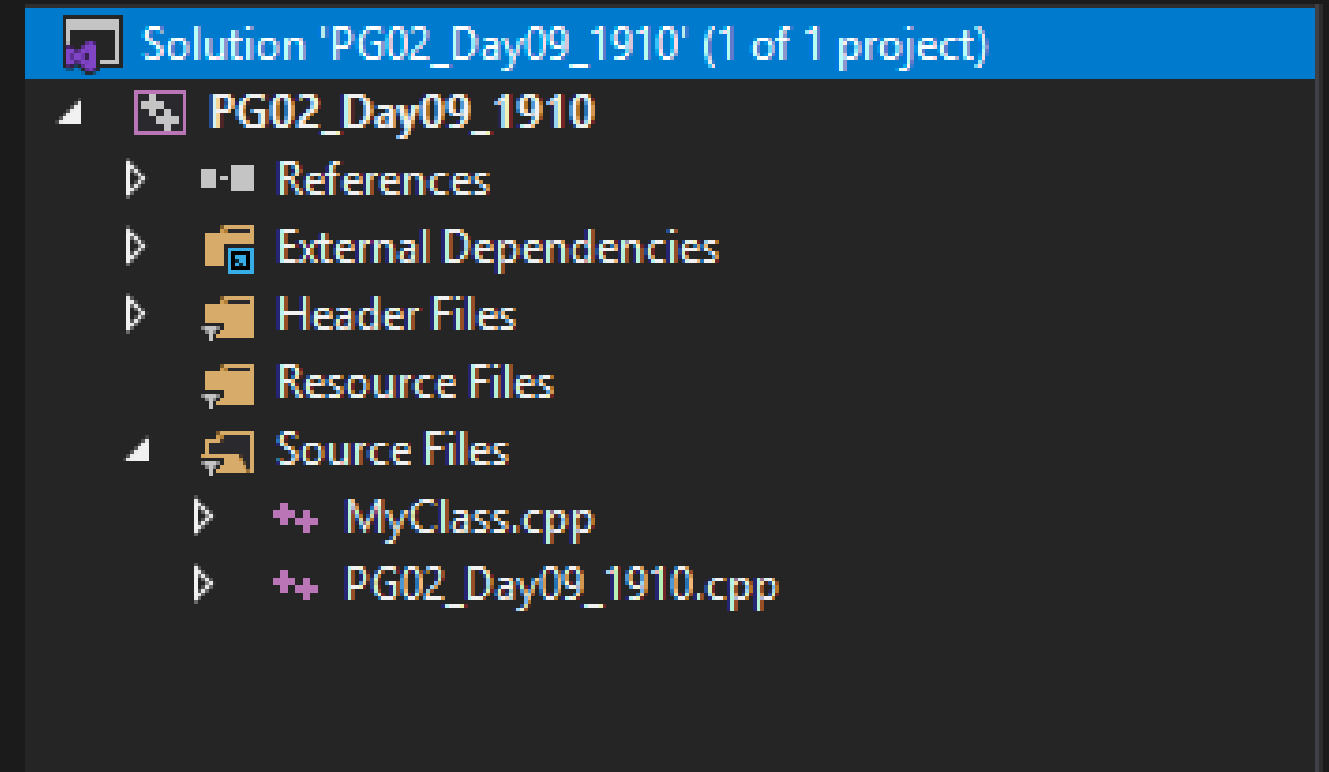
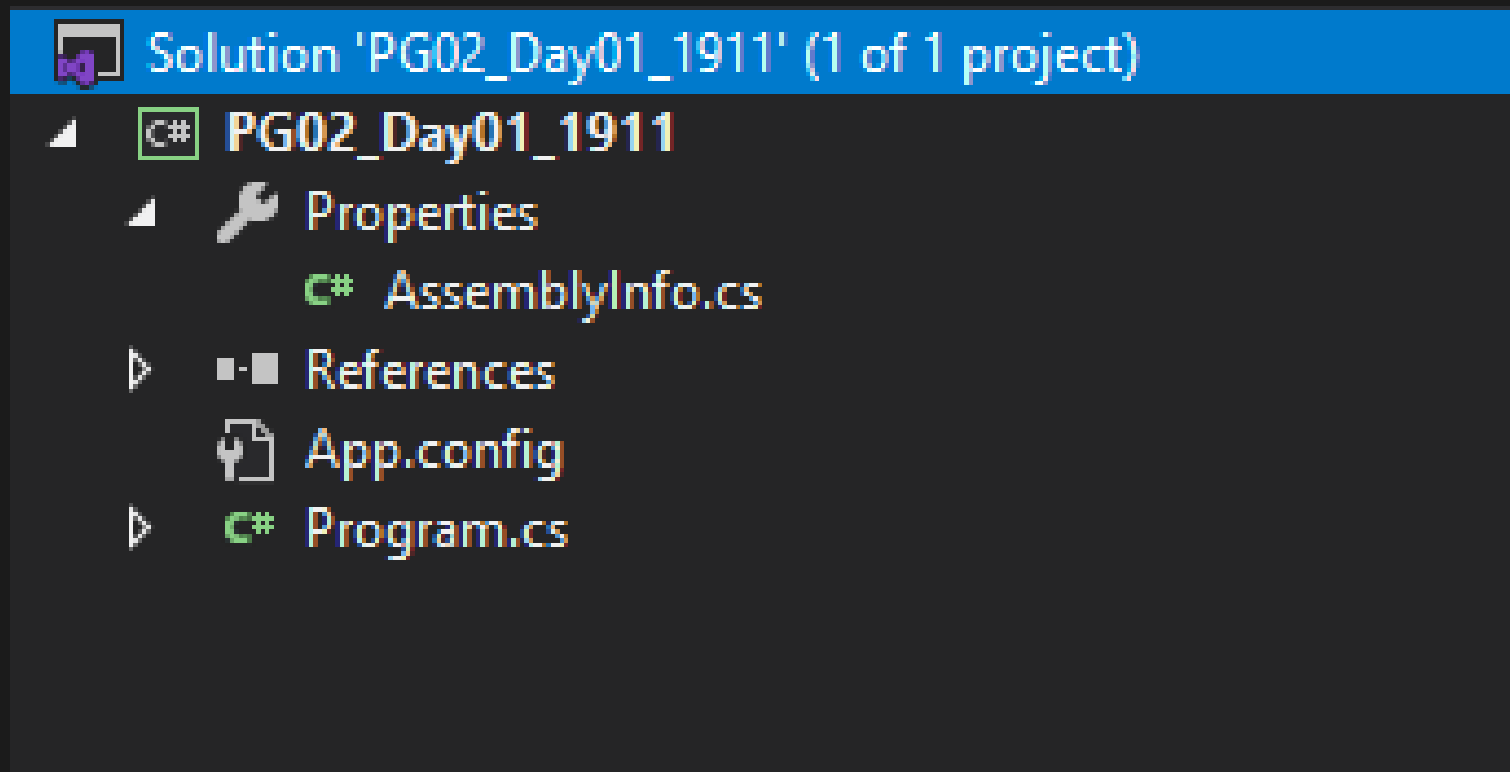
Part 5-1: !Hello World

C++

- Let's create a **C++ Windows Console Application**
- Visual Studio 2019

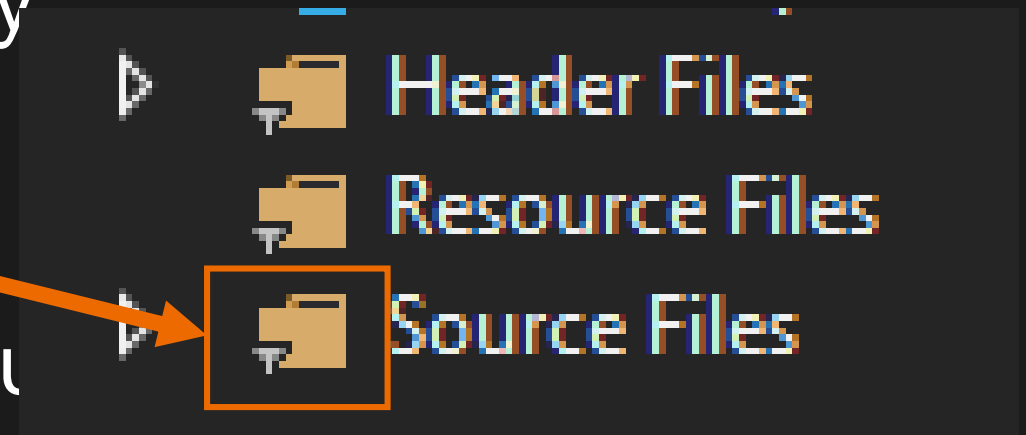


C# vs C++



C++

- Let's explore the files and folder structure
 - Look at the location of the generated **pch.h**. Even though it is under a Header Files folder, it is actually stored in the same place as the other files. The 'folders' in solution explorer are actually filters.
 - Where files are on disk is where they stay when you add them to the project. It stores a relative path in the project file.
 - C# would make a copy of the file and place them under the folder on disk.



C++

C#

```
using System;
```

```
namespace ConsoleApp2
```

```
{
```

```
    0 references
```

```
    class Program
```

```
    {
```

```
        0 references
```

```
        static void Main(string[] args)
```

```
        {
```

```
            Console.WriteLine("Hello World");
```

```
        }
```

```
    }
```

```
}
```

```
#include <iostream>
```

```
int main(int argc, char** argv)
```

```
{
```

```
    std::cout << "Hello World!\n";
```

```
}
```


Exercises

For this lecture, it would be best to open the [lab document](#) and work on the sections as we go through the lecture.

Exercise: Part 5-1

1. Create a new **C++ console application**
2. Print out any message you want EXCEPT “Hello World”

Part 5-2: Variables

Variables

TYPE	C#	C++
int	int number = 5;	int number = 5;
bool	bool isPlaying = true;	bool isPlaying = true;
float	float range = 5.3F;	float range = 5.3F;
double	double depth = 3457.6;	double depth = 3457.6;
char	char symbol = 'B';	char symbol = 'B';

Arrays

C#

- `int[] numbers = new int[5]{1,2,3,4,5};` (**heap** array)

C++

- `int numbers[5] = {1,2,3,4,5};` (**stack** array)
- `int* numbers = new int[5]{1,2,3,4,5};` (**heap** array)

Exercise: Part 5-2

1. Create variables of the following types: int, bool, float, double, char.
2. Print the variables and the `sizeof` the variables using `cout`.
3. Create an array of floats. Print the array.
4. Create an array of char with "Batman". Print the most awesome char array.

Part 5-3: Loops

for Loops

C#

- `for (int i = 0; i < length; i++)`

C++

- `for (int i = 0; i < length; i++)`

while Loops

C#

- `while (!exit)`

C++

- `while (!exit)`

do-while Loops

C#

```
do  
{  
} while (!exit);
```

C++

```
do  
{  
} while (!exit);
```

Exercise: Part 5-3

1. Write a **for loop** from 0 to 100 (inclusive). Print out only the even numbers.
2. Write a **while loop** that loops until a counter reaches 100.
Print the counter in the loop.
3. Write a **do-while loop** that loops until a counter reaches 100.
Print the counter in the loop.

Part 5-4: Ifs, Random Numbers

Random Numbers

C#

```
Random randy = new Random((int)DateTime.Now.Ticks);
```

```
int num = randy.Next();
```

```
num = randy.Next(101);
```

```
num = randy.Next(5, 250);
```

C++

```
srand((unsigned)time(NULL));
```

```
int num = rand();
```

if, if-else, if-else-if

C#

```
if(value < 25)
    Console.WriteLine("Quad 1");
else if(value < 50)
    Console.WriteLine("Quad 2");
else if (value < 75)
    Console.WriteLine("Quad 3");
else
    Console.WriteLine("Quad 4");
```

C++

```
if (value < 25)
    cout << "Quad 1";
else if (value < 50)
    cout << "Quad 2";
else if (value < 75)
    cout << "Quad 3";
else
    cout << "Quad 4";
```

Exercise: Part 5-4

1. Generate 10 random numbers from 0 to 100 (inclusive).
2. Print out the number AND the associated letter grade. Use if-elseif.

Part 5-5: Switch

Switch

C#

```
switch (value)
{
    case 1:
        break;
    case 2:
        break;
    case 3:
    case 4:
        break;
    default:
        break;
}
```

C++

```
switch (value)
{
    case 1:
        break;
    case 2:
        break;
    case 3:
    case 4:
        break;
    default:
        break;
}
```

Switch - differences

C#

```
switch (value)
{
    case 1:
        Console.WriteLine("1");
    case 2:
        Console.WriteLine("2");
        break;
}
```

C++

```
switch (value)
{
    case 1:
        cout << "1";
    case 2:
        cout << "2";
        break;
}
```

Exercise: Part 5-5

1. Generate 10 random numbers from 0 to 5 (inclusive).
2. Print out the result:
 - 0: "The Bat"
 - 1: "Batman"
 - 2: "Bats"
 - 3: "The Dark Knight"
 - 4: "Nightwing"
 - 5: "Bruce"
3. DO NOT USE if.

Part 5-6: Enums

Enums

C#

```
enum ShipClassification
{
    Cruiser,
    Frigate,
    Battleship,
    Destroyer,
    Carrier,
    Tug
}
```

C++

```
enum ShipClassification
{
    Cruiser,
    Frigate,
    Battleship,
    Destroyer,
    Carrier,
    Tug
};
```

Can you spot the differences?

Exercise: Part 5-6

1. Create an enum for 5 superpowers
2. Randomly pick one of them. HINT: you'll have to cast a random number to the superpower type.
3. DO NOT USE if.
4. Create an enum called Comp with the following values: -1, 0, 1 called Less, Equal, Greater.
5. Call the strcmp method with "Batman" and "Aquaman".
6. Cast the result to a Comp variable.

Part 5-7: Methods

Global

- The Global namespace can hold variables and methods.
- To use global methods in your code, the method has to be defined before your code.

Methods

- Methods in C++ are declared like C# methods

- Return type

- Name of method

- List of parameters



int Add(int num1, int num2)

{

 return num1 + num2;

}

Exercise: Part 5-7

1. Write a Factorial method in the global namespace above main.
2. It should take 1 int parameter and return a long.
3. In main, write a for loop to call and print the factorial result using the for loop variable as the parameter to the method.

Part 5-8: Vectors

vectors

- List<T> in C# is the class that is an expandable array.
- In C++, that class is called **vector<Type>**
- Vector is part of the std namespace.
- To use, you need to **#include <vector>**

vectors

- To **declare** a vector variable:
 - `std::vector<int> grades;`
- To **add items** to a vector:
 - `grades.push_back(5);`
- To **remove items** from a vector:
 - `grades.pop_back();`
- To **access** an item in the vector:
 - `cout << grades[0];`
- To get the **number of elements** in a vector:
 - `cout << "Size: " << grades.size();`

Exercise: Part 5-8

1. Create a vector of char named `letterGrades` in the global namespace.
2. Write a `Grade` method in the global namespace. It should take 1 int parameter for the numeric grade and should return the associated letter grade from the vector.
3. In main, generate 10 random grades from 0-100 inclusive. Call the `Grade` method passing each numeric grade. Print the letter grade returned from the method.

For More Info

- XOR operator
<https://msdn.microsoft.com/en-us/library/3akey979.aspx>
- C++
<http://www.cplusplus.com/doc/tutorial/>
- Pointers
<http://www.cplusplus.com/doc/tutorial/pointers/>