Principle of Programing

1. Computer programming concepts

Lecturers:

Sipan M. Hameed

www.sipan.dev

2024 - 2025

1. Table of Contents

2

1. Computer programming concepts

1.1 Computer program

A computer program is a collection of data and instructions that can be executed by a computer to perform a specific task. A computer program is usually written by a computer programmer in a programming language.

1.2 computer programming

Computer programming is a way of giving computers instructions about what they should do next. These instructions are known as code, and computer programmers write code to solve problems or perform a task.

1.3 programming language

A programming language is a computer language programmer use to develop software programs, scripts, or other sets of instructions for computers to execute.

1.4 Popular Programming Languages

The TIOBE Programming Community index is an indicator of the popularity of programming languages. The index is updated once a month. The ratings are based on the number of skilled engineers world-wide, courses and third-party vendors. Popular search engines such as Google, Bing, Yahoo!, Wikipedia, Amazon, YouTube and Baidu are used to calculate the ratings. It is important to note that the TIOBE index is not about the best programming language or the language in which most lines of code have been written.

Feb 2021	Feb 2020	Change	Programming Language	Ratings	Change
1	2	⌃	C	16.34%	-0.43%
2	1	⌄	Java	11.29%	-6.07%
3	3		Python	10.86%	+1.52%
4	4		C++	6.88%	+0.71%
5	5		C#	4.44%	-1.48%
6	6		Visual Basic	4.33%	-1.53%
7	7		JavaScript	2.27%	+0.21%
8	8		PHP	1.75%	-0.27%
9	9		SQL	1.72%	+0.20%
10	12	⌃	Assembly language	1.65%	+0.54%

1.5 Computer programming concepts

The concepts programming is essential knowledge to anyone who wants to become skilled in computer programming. While some are not universal, these concepts are present in the majority of computer programming languages and/or are a fundamental part of the programming process.

1.5.1 Algorithm

A set of steps for carrying out a specific task. Algorithms are used extensively in computer programming to arrive at a solution for a problem. The process of creating an algorithm involves documenting all the necessary steps needed to arrive at the solution and how to perform each step. A real-world example of an algorithm would be a recipe. The instructions of a typical recipe (add ingredients, mix, stir, etc.) are an algorithm.

1.5.2 Source code

The actual text used to write the instructions for a computer program. This text is then translated into something meaningful the computer can understand.

1.5.3 Compiler

A software tool that translates source code into data that the computer can understand. Specifically, a compiler is used to turn source code into object code. The object code is then passed through a program called a linker which turns it into an executable program.

1.5.4 Data type

The classification of pieces of information in a program. The amount of different data types varies between languages. Typically, there are data types for integers (whole numbers), floating-point numbers (numbers with a decimal part), and single characters. To distinguish between different data types, a computer uses special internal codes.

1.5.5 Variable

A container which represents a value in a program. Variables can store different types of data including numeric values, single characters, and text strings. The value of a variable can change all throughout a program.

1.5.6   Constant

The same thing as a variable with one major difference - the value of a constant does not change, while the value of a variable can change all throughout a program.

1.5.7   Conditional

A set of code that will execute only if a certain condition is true. Conditionals are used to test expressions and perform certain operations accordingly. For example, you could test a number input by the user and if it is too high print the message "The number entered is too high" and the program exits. Thanks to conditionals, a program can work differently every time it runs.

1.5.8   Array

A special type of variable used in many programming and web languages including PHP, JavaScript, and Java that contains a list of related values. For example, a colors array would contain a list of colors.

1.5.9   Loop

A segment of code that executes repeatedly based on a certain condition. Loops are used to perform tasks repeatedly a certain number of times. For example, if you needed to print the numbers 1 to 10. You can use a loop for this task instead of manually printing all the numbers.

1.5.10   Function

A set of code used to carry out specific tasks. A function can take parameters which will affect its output as well as return values. Functions prevent unnecessary redundancy because you can use them as much as needed instead of retyping some code over and over. For example, if you need to multiply two numbers, instead of doing the calculation manually every time, you can supply the data to a function through some parameters which will do it for you.

1.5.11   Class

A template for a real-world object to be used in a program. For example, a programmer can create a car class which represents a car. This class can contain the properties of a car (color, model, year, etc.) and functions that specify what the car does (drive, reverse, stop, etc.). Classes are used in object-oriented programming.

1.6 C#

C# is pronounced "C-Sharp". It is an object-oriented programming language created by Microsoft that runs on the .NET Framework. C# has roots from the C family, and the language is close to other popular languages like C++ and Java.

The first version was released in year 2002. The latest version, C# 8, was released in September 2019.

C# is used for:

- Mobile applications
- Desktop applications
- Web applications
- Web services
- Web sites
- Games
- VR
- Database applications
- And much, much more!

1.6.1 C# IDE

The easiest way to get started with C#, is to use an IDE. An IDE (Integrated Development Environment) is used to edit and compile code.

In our tutorial, we will use Visual Studio Community, which is free to download from https://visualstudio.microsoft.com/vs/community/.

Applications written in C# use the .NET Framework, so it makes sense to use Visual Studio, as the program, the framework, and the language, are all created by Microsoft.
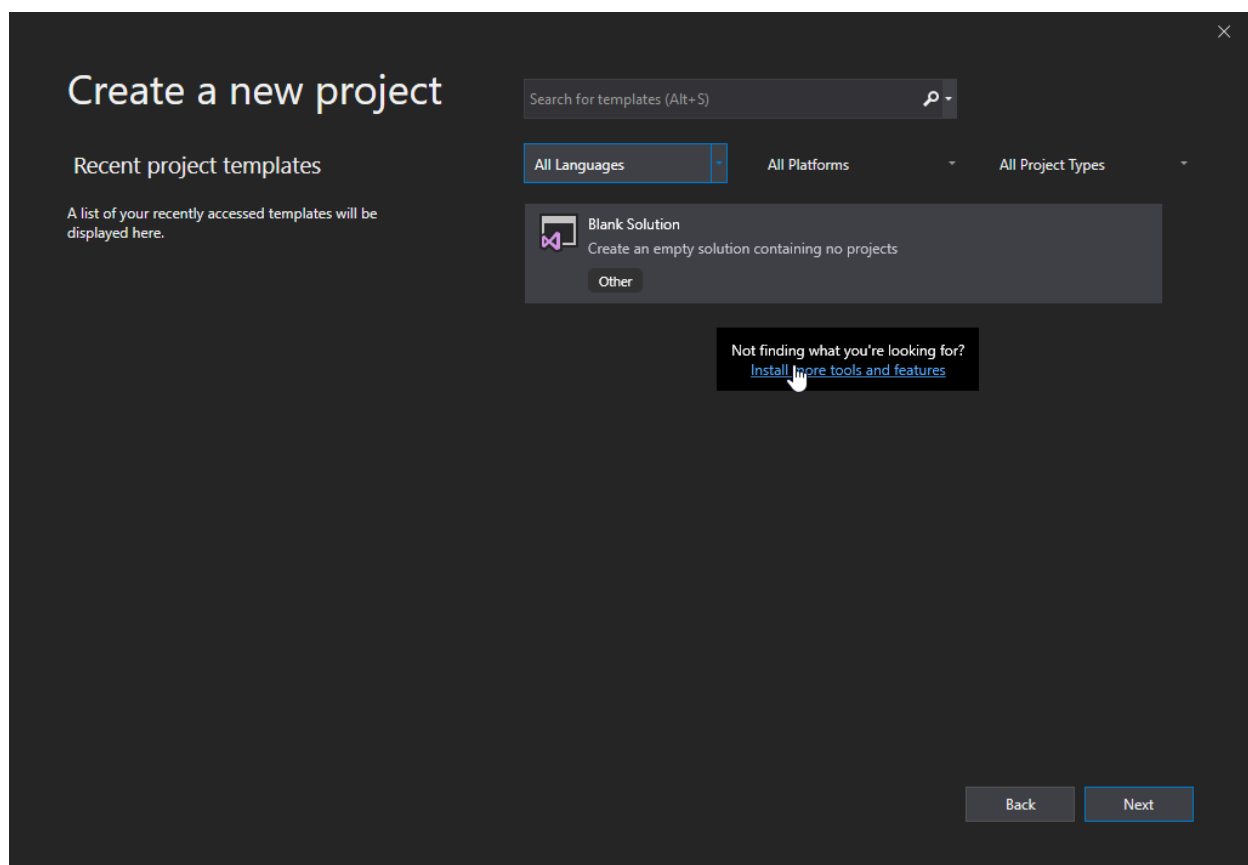
1.6.2 C# Install

Once the Visual Studio Installer is downloaded and installed, choose the .NET workload and click on the Modify/Install button:

After the installation is complete, click on the **Launch** button to get started with Visual Studio.

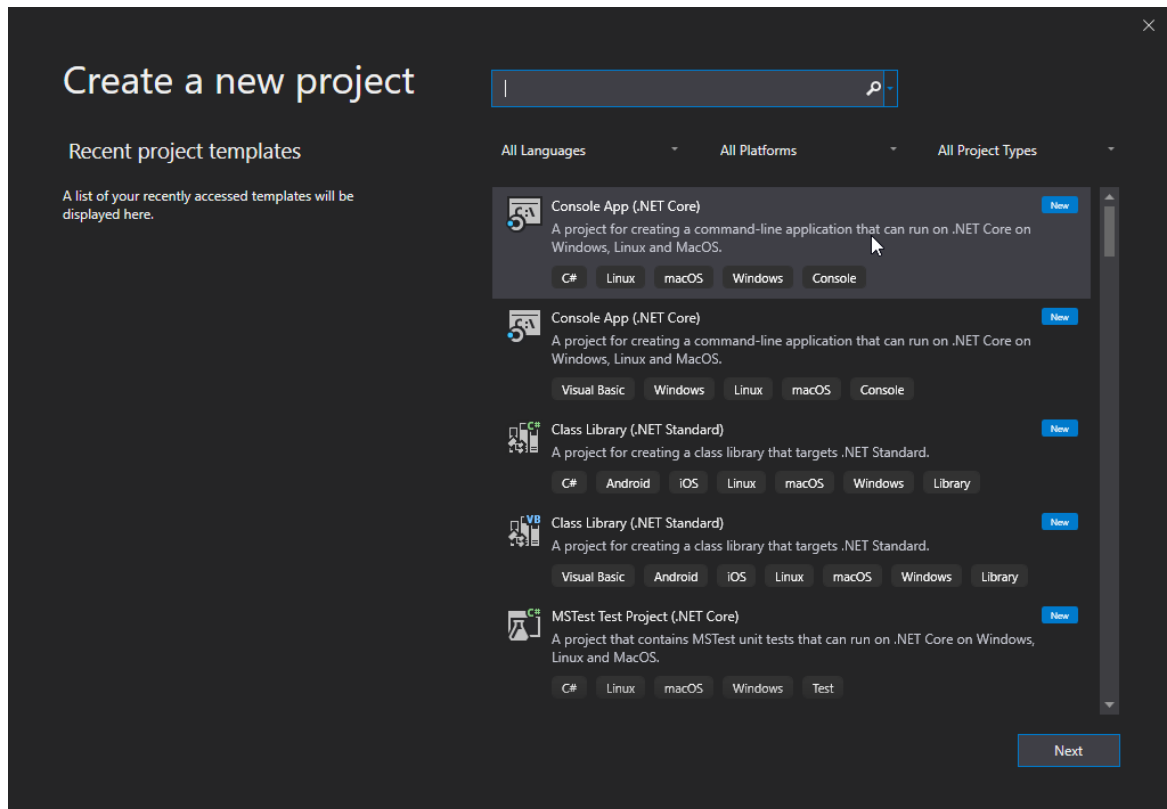On the start window, choose **Create a new project**:
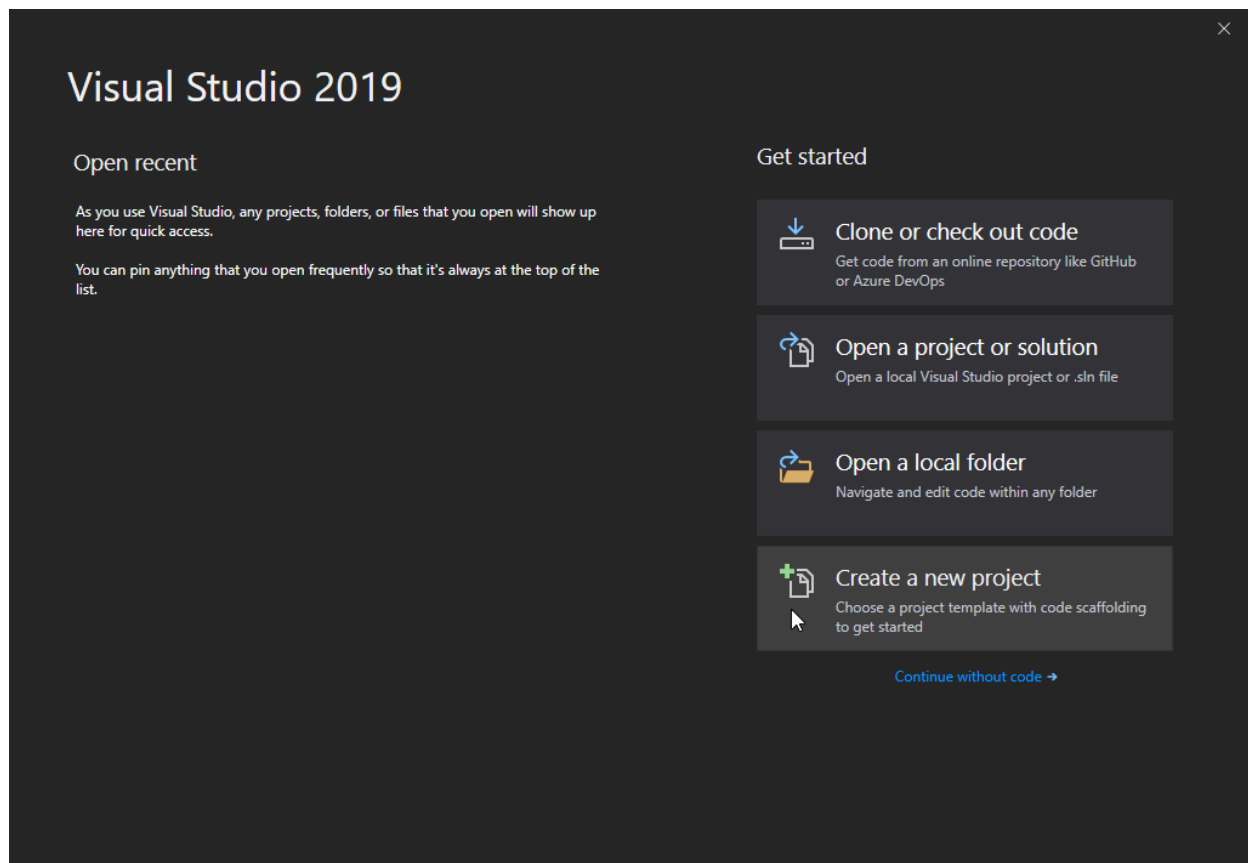
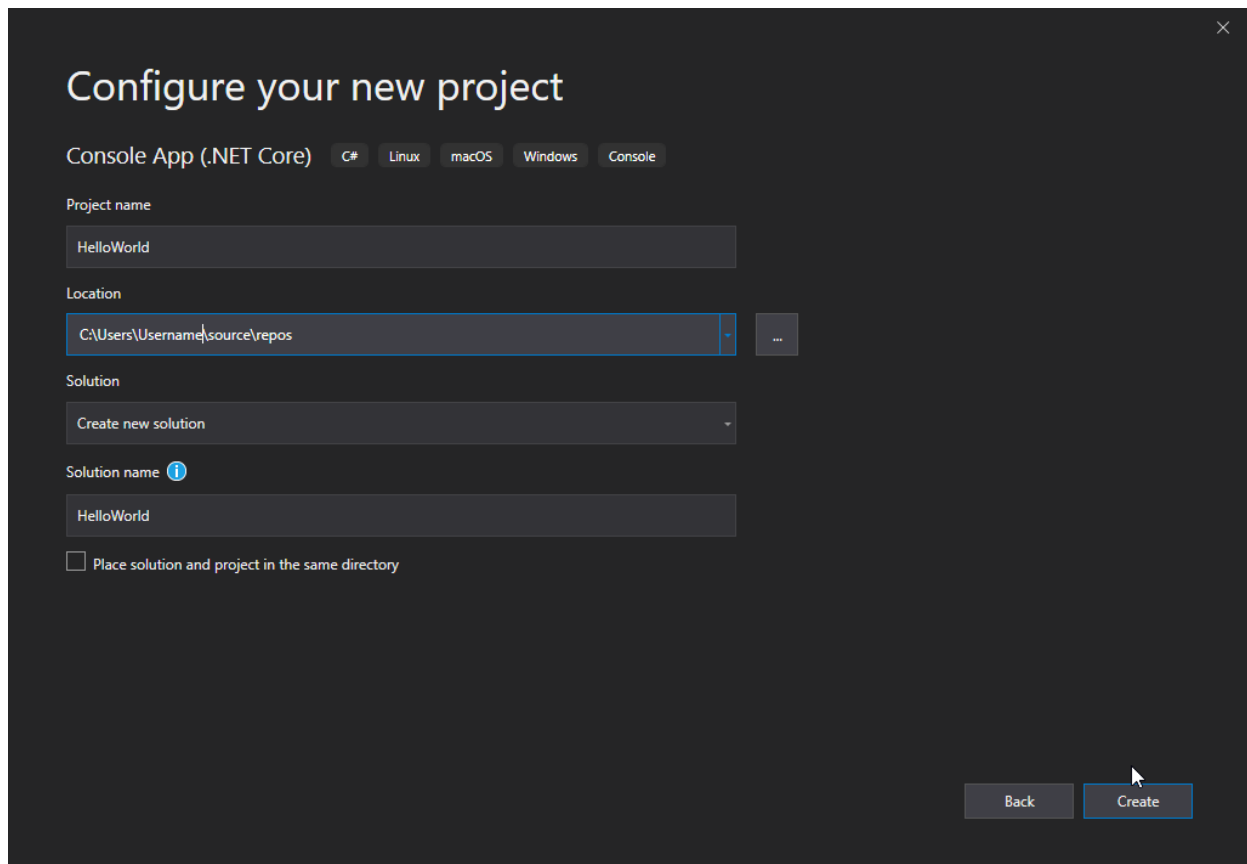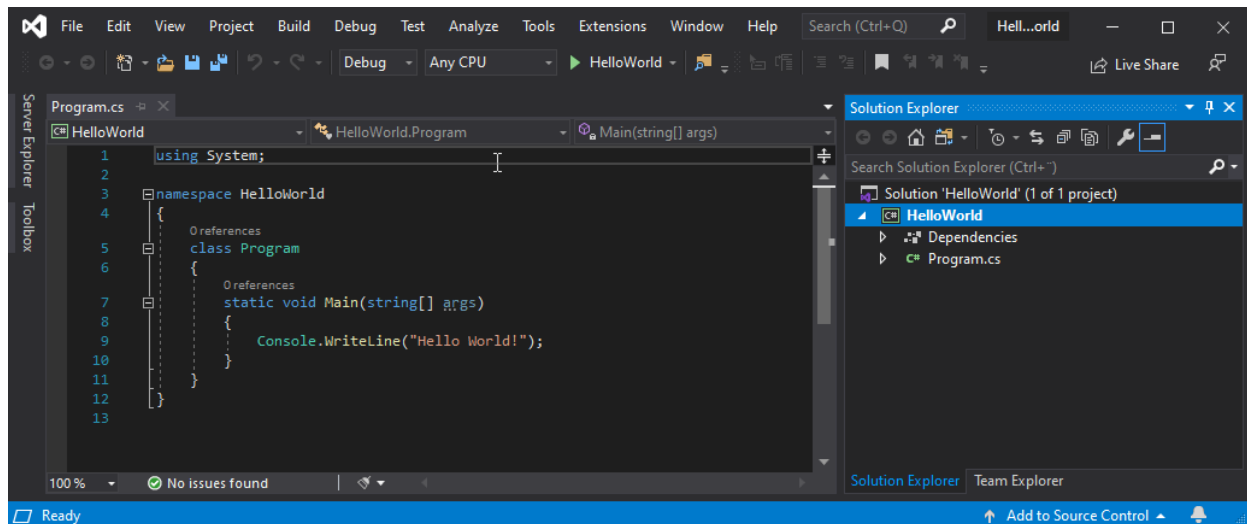Then click on the "Install more tools and features" button:

Choose "Console App (.NET Core)" from the list and click on the Next button:

Enter a name for your project, and click on the Create button:

Visual Studio will automatically generate some code for your project:

The code should look something like this:

Program.cs

```
using System;

namespace HelloWorld
{
  class Program
  {
    static void Main(string[] args)
    {
      Console.WriteLine("Hello World!");
    }
  }
}
```

1.6.3 C# Syntax

In the previous chapter, we created a C# file called Program.cs, and we used the following code to print "Hello World" to the screen:

Program.cs

```
using System;

namespace HelloWorld
{
  class Program
  {
    static void Main(string[] args)
    {
      Console.WriteLine("Hello World!");
    }
  }
}
```

1.7 Practical Examples

1.7.1 C# Sharp Basic: Exercise-1 with Solution

Write a C# Sharp program to print Hello and your name in a separate line.

Sample Solution:

C# Sharp Code:

```csharp
public class Exercise1
{
    public static void Main( )
    {
        System.Console.WriteLine("Hello");
        System.Console.WriteLine("Zaxo IT Student!");
    }
}
```

Sample Output:

```
Hello
Zaxo IT Student!
```

1.7.2 C# Sharp Basic: Exercise-2 with Solution

Write a C# Sharp program to print the sum of two numbers.

In mathematics, summation (capital Greek sigma symbol: $\sum$) is the addition of a sequence of numbers; the result is their sum or total. The numbers to be summed may be integers, rational numbers, real numbers, or complex numbers.

C# sharp Exercises: Print the sum of two numbers

Sample Solution:C# Sharp Code:

```csharp
public class Exercise2
{
    public static void Main()
    {
        System.Console.WriteLine(15+17);
    }
}
```

Sample Output:

```
32
```

1.7.3 C# Sharp Basic: Exercise-3 with Solution

Write a C# Sharp program to print the result of dividing two numbers. Division is one of the four basic operations of arithmetic, the others being addition, subtraction, and multiplication. The division of two natural numbers is the process of calculating the number of times one number is contained within one another.

C# sharp Exercises: dividing two numbers

Sample Solution: C# Sharp Code:

```
public class Exercise3
{
  public static void Main()
  {
    System.Console.WriteLine(36/6);
  }
}
```

Sample Output:

```
6
```

1.7.4 C# Sharp Basic: Working with Arithmetic Expression

Write a C# Sharp program to print the result of the specified operations.

C# Sharp: Working with Arithmetic Expression

Sample Solution:

C# Sharp Code:

```
public class Exercise4
{
    public static void Main()
    {
        System.Console.WriteLine(-1+4*6);

        System.Console.WriteLine((35+5)%7);

        System.Console.WriteLine(14+-4*6/11);

        System.Console.WriteLine(2+15/6*1-7%2);

    }
}
```
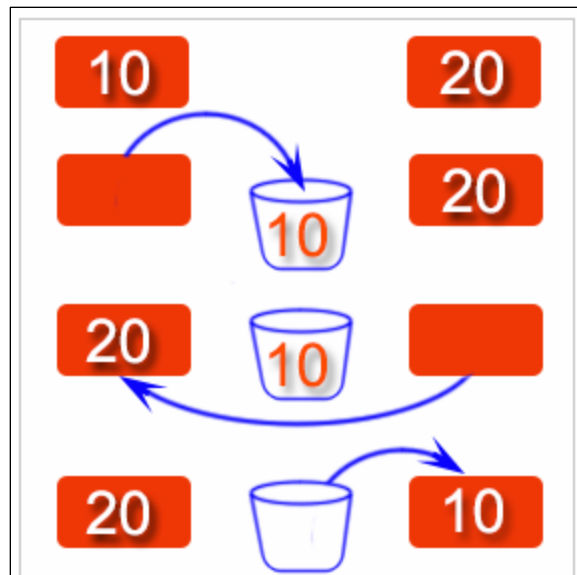
Sample Output:

```
23
5
12
3
```

1.7.5 C# Sharp Basic: swapping two variables

Write a C# Sharp program to swap two numbers. **C# Sharp: swapping two variables**

The act of swapping two variables refers to mutually exchanging the values of the variables. Generally, this is done with the data in memory. Using a temporary variable : The simplest method to swap two variables is to use a third temporary variable :

Sample Solution: C# Sharp Code:

```csharp
using System;
public class Exercise5
{
      public static void Main(string[] args)
        {
            int number1, number2, temp;
            Console.Write("\nInput the First Number : ");
            number1 = int.Parse(Console.ReadLine());
            Console.Write("\nInput the Second Number : ");
            number2 = int.Parse(Console.ReadLine());
            temp = number1;
            number1 = number2;
            number2 = temp;
            Console.Write("\nAfter Swapping : ");
            Console.Write("\nFirst Number : "+number1);
            Console.Write("\nSecond Number : "+number2);
            Console.Read();
        }
}
```

Sample Output:

```
Input the First Number : 2
Input the Second Number : 5
After Swapping :
First Number : 5
Second Number : 2
```

1.7.6 C# Sharp Basic: Exercise-6 with Solution

Write a C# Sharp program to print the output of multiplication of three numbers which will be entered by the user. **C# Sharp multiplication (*) operator:**

The multiplication operator (*), which computes the product of its operands. Also, the dereference operator, which allows reading and writing to a pointer.

Note:

- All numeric types have predefined multiplication operators.
- The * operator is also used to declare pointer types and to dereference pointers.
- User-defined types can overload the binary * operator. When a binary operator is overloaded, the corresponding assignment operator, if any, is also implicitly overloaded.

Sample Solution:

C# Sharp Code:

```
using System;
public class Exercise6
{
  public static void Main()
  {
    int num1, num2, num3;

    Console.Write("Input num1: ");
    num1 = Convert.ToInt32(Console.ReadLine());

    Console.Write("Input nym2: ");
    num2 = Convert.ToInt32(Console.ReadLine());

    Console.Write("Input num3: ");
    num3 = Convert.ToInt32(Console.ReadLine());

    int result = num1 * num2 * num3;
    Console.WriteLine("Output: {0} x {1} x {2} = {3}",
                      num1, num2, num3, result);
  }
}
```
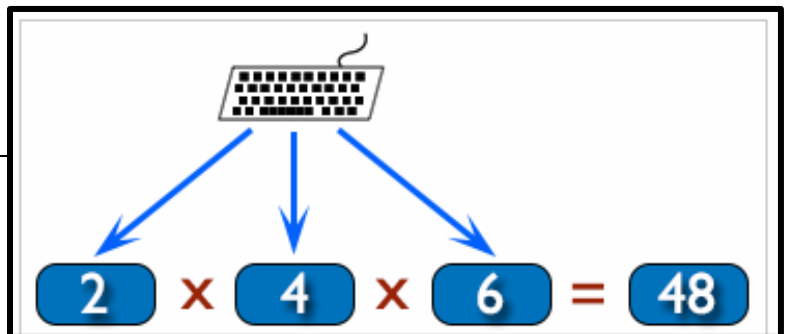
Sample Output:

```
Input num1: 2
Input num2: 8
Input num3: 5
Output: 2 x 8 x 5 = 80
```