Lập Trình C# - SoloLearn.com

**Student: GreenWolf Date: 23/12/2020**

# Basic Concepts

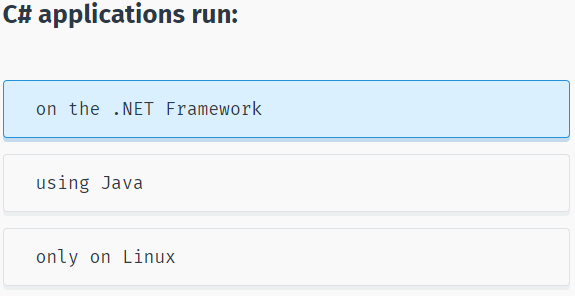
## What is C#?

**Welcome to C#**

C# is an elegant object-oriented language that enables developers to build a variety of secure and robust applications that run on the **.NET Framework**.

You can use C# to create Windows applications, Web services, mobile applications, client-server applications, database applications, and much, much more.

! You will learn more about these concepts in the upcoming lessons!



**The .NET Framework**

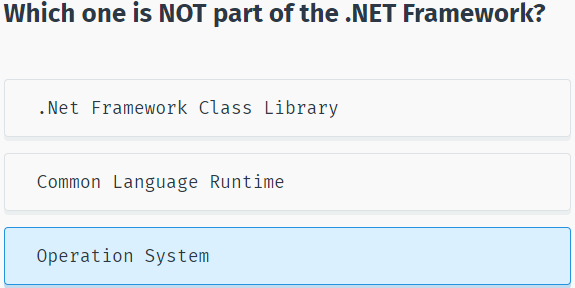
The .NET Framework consists of the **Common Language Runtime (CLR)** and the .NET Framework **class library**.

The **CLR** is the foundation of the .NET Framework. It manages code at execution time, providing core services such as memory management, code accuracy, and many other aspects of your code.

The **class library** is a collection of classes, interfaces, and value types that enable you to accomplish a range of common programming tasks, such as data collection, file access, and working with text.

C# programs use the .NET Framework class library extensively to do common tasks and provide various functionalities.

! These concepts might seem complex, but for now just remember that applications written in C# use the .NET Framework and its components.



## Variables

**Variables**

Programs typically use data to perform tasks.

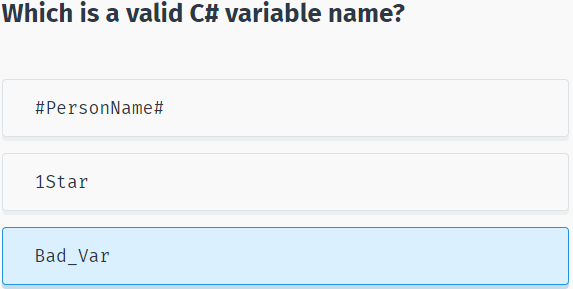
Creating a **variable** reserves a memory location, or a space in memory, for storing values. It is called **variable** because the information stored in that location can be changed when the program is running.

To use a variable, it must first be declared by specifying the **name** and **data type**.

A variable name, also called an **identifier**, can contain letters, numbers and the underscore character (\_) and must start with a letter or underscore.

Although the name of a variable can be any set of letters and numbers, the best identifier is **descriptive** of the data it will contain. This is very important in order to create clear, understandable and readable code!

! For example, firstName and lastName are good descriptive variable names, while abc and xyz are not.



**Variable Types**

A **data type** defines the information that can be stored in a variable, the size of needed memory and the operations that can be performed with the variable.

For example, to store an integer value (a whole number) in a variable, use the **int** keyword:

int myAge;

The code above declares a variable named **myAge** of type **integer**.

!A line of code that completes an action is called a statement. Each statement in C# must end with a **semicolon ‘;’**.

You can assign the value of a variable when you declare it:

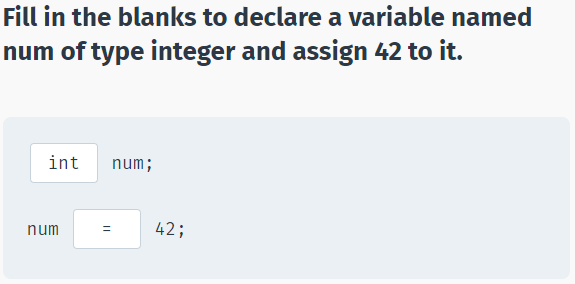
int myAge = 18;

or later in your code:

int myAge;

myAge = 18;

!Remember that you need to declare the variable before using it.



**Built-in Data Types**

There are a number of built-in data types in C#. The most common are:

**int** - integer.

**float** - floating point number.

**double** - double-precision version of float.

**char** - a single character.

**bool** - Boolean that can have only one of two values: True or False.

**string** - a sequence of characters.

The statements below use C# data types:

int ​x = 42;

double ​pi = 3.14

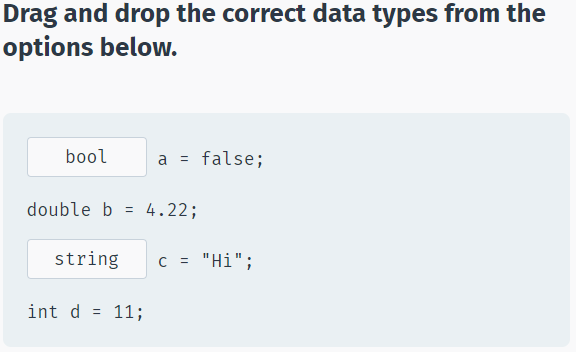
char y = 'Z'

bool ​isOnline = true;

string ​firstName = "David”;

!Note that char values are assigned using single quotes and string values require double quotes.

You will learn how to perform different operations with variables in the upcoming lessons!



## Your First C# Program

Your First C# Program

You can run, save, and share your C# codes on our Code Playground, without installing any additional software.

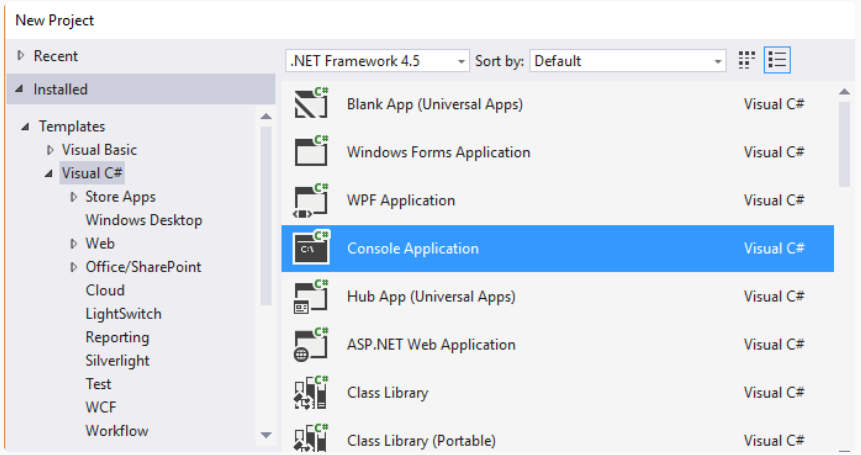
!Reference this lesson if you need to install the software on your computer.

To create a C# program, you need to install an integrated development environment (IDE) with coding and debugging tools.

We will be using **Visual Studio Community Edition**, which is available to download for free.

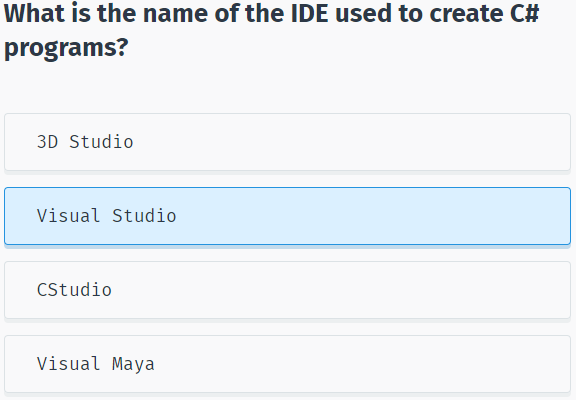
After installing it, choose the default configuration.

Next, click **File->New->Project** and then choose **Console Application** as shown below:



Enter a name for your Project and click OK.

Console application uses a text-only interface. We chose this type of application to focus on learning the fundamentals of C#.



# Conditionals and Loops

# Methods

# Classes & Objects

# Arrays & Strings

# More On Classes

# Inheritance & Polymorphism

# Inheritance & Polymorphism

# Generics

# Dictionary

Basic Concepts: Các khái niệm cơ bản

data types: Các kiểu dữ liệu

arrays : Mảng

pointers: Con trỏ

conditional statements: Câu lệnh điều kiện

loops: Vòng lặp

functions: Hàm

classes: lớp

objects: đối tượng

inheritance: Kế thừa

polymorphism: Đa hình

Basic Arithmetic: Số Học cơ bản

Assignment: Phân Công

Increment: Gia Tăng

decrement: giảm đi

cross-platform language: Ngôn ngữ đa nền tảng

popular: phổ biến

high-performance: Hiệu suất cao

applications: ứng dụng

operating systems: hệ điều hành

browsers: trình duyệt

video-games: trò chơi điện tử

programming language: ngôn ngữ lập trình

General purpose: Mục đích chung

template: bản mẫu

commands: câu lệnh

statements: Mệnh đề

irrespective of: ko phân biệt (bất kể)

entry point: Đầu vào

input: Nhập Liệu đầu vào

output: xuất hình đầu ra

streams: luồng xủ lý

combination: sự kết hợp

standard: tiêu chuẩn thông thường

display screen: màn hình hiển thị

terminate: chấm dứt

instruction: chỉ dẫn

offer: cung cấp

various headers: tiêu đề khác nhau

work properly: hoạt động đúng cách

namespace:không gian tên

declarative region: vùng khai báo

features: các tính chất

insert: thêm vào

Variables: Biến

memory location: vị trí bộ nhớ

reserves: dự trữ

declare: khai báo

syntax error: lỗi cú pháp

Specify: chỉ định

Addition: phép cộng

Subtraction: Phép Trừ

Multiplication: Phép Nhân

Division: Phép Chia lấy nguyên

Modulus: Phép Chia Lấy Dư

Parentheses: dấu ngoặc đơn

innermost: trong cùng

evaluate: đánh giá

expressions nested: biểu thức lồng nhau

quadratic equation: Phương Trình Bậc 2

intended to: mục đích

Expressions: biểu thức

legal: hợp pháp

illegal: bất hợp pháp

positive numbers: số dương

Case-Sensitivity: phân biệt chữ hoa chữ thường

meaningful: có ý nghĩa

practical: thiết thực

occasionally: thỉnh thoảng

necessary: cần thiết

elements: nhiều phần tử

rows: hàng

columns: cột

ampersand: dấu &

prototypes: nguyên mẫu

declaration: khai báo

arguments: đối số

parameters: tham số

result: kết quả

Overloading: quá tải

Reference: tham chiếu

identity: bản sắc, danh tính...

attributes: thuộc tính

behavior: hành vi

In regard to: liên quan đến

instantiation: sự tức thời (sự khởi tạo đối tượng )

Abstraction: Tính trừu tượng

.NET Framework: Môi trường làm việc với .net