Mr. Bellavia

# LESSON 04 - PROGRAMMING A COMPUTER (SOFTWARE)

In this lesson we will explore the different types of software (programs) that a computer generally runs. We will also introduce the concept of computer programing (i.e., writing/creating computer software). The previous lessons examined the basic hardware of a computer; however, the primary focus of Computer Science is the study of computer programming.

**Sections:**



## I. TYPES OF SOFTWARE:

**Definition of Software:** Software (also referred to as computer programs) is a collection of instructions that tell a computer what to do. Whereas hardware is physical (i.e., any component you can see or touch in a computer), software is virtual (it does not exist in the physical world and is purely conceptual).

Software that runs on your computer comes in two basic forms:

### 1) System Software:

System software basically runs your computer. It controls all the hardware inside your computer, helps launch and run other software and allows interaction with the user.

When you start your computer, it is a system software that is initialized. This type of system software is referred to as an **Operating System (OS).**

For desktops and laptops, some examples of operating systems are:

- Windows
- MAC OS (Apple)
- Linux

For mobile devices like phones and tablets, some examples of operating systems are:

- Apple iOS
- Google Android

These are the most popular operating systems out there, but there are many others that do exist.

Other types of system software include **device drivers** and **utilities.** A **device driver** is software that controls a device attached to a computer, for example a printer driver for a

printer. A **utility** is software designed to assist users in the maintenance and care of their computers, for example a virus scanner.

**2) Application Software:**

Application software (apps) are programs that most computer users use to perform certain tasks. The list of available computer applications is virtually endless. Here are some examples:

- Word Processing Software - Used to type documents, for example Microsoft Word.
- Photo Editing Software – Used to edit photos, for example Adobe Photoshop.
- Internet Browsers – Used to browse websites on the Internet, for example Google Chrome.
- And the list goes on and on…

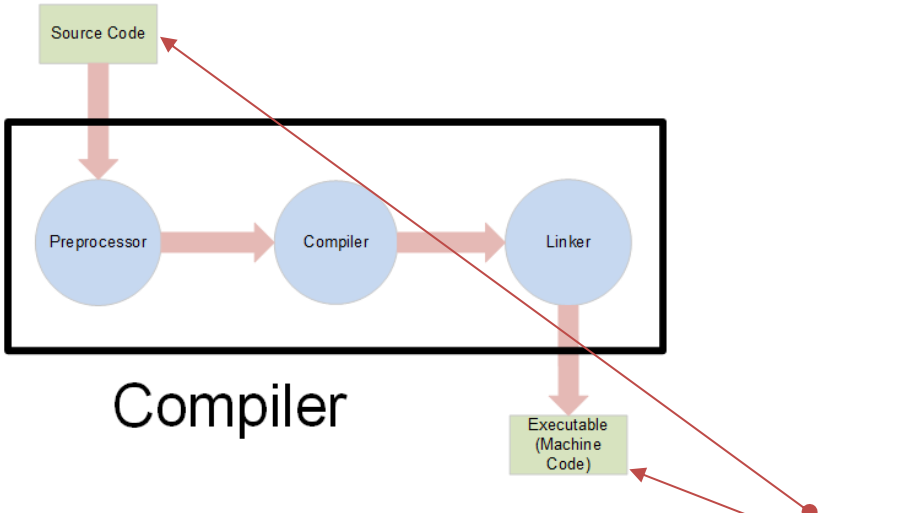
Another big type of application software is what we will use in this course which is called **Programming Software.** This type of software is used by computer programmers who wish to write/code their own software. In this course we will explore programming software so that we can write our own programs!

## II. COMPUTER PROGRAMMING LANGUAGES:

In this course you will learn how to program a computer (i.e., write/code your own software) using a **computer programming language.** Just like we communicate with each other using the English language, we communicate with a computer using a computer programming language **(computer code)**. The programming language we will use in this course is called **C# (C-Sharp).** C# has been around for a while and has been constantly improved over the years.

Back when computers were first being developed, computers were programmed using **machine language** which consists of binary instructions sent to the CPU. This was very tedious work, and it is virtually impossible to write todays software using just 0s and 1s. Machine languages are refereed to as **low-level languages** since they work at the hardware level.

Luckly today we have what are called **high-level languages,** like **C#**, which consists of English like words and makes programming a lot easier. When you write a program in C# it will be transformed into machine language by a **compiler.** We do not need to get into the details of how a compiler works, but here is a quick diagram to demonstrate what is going on:

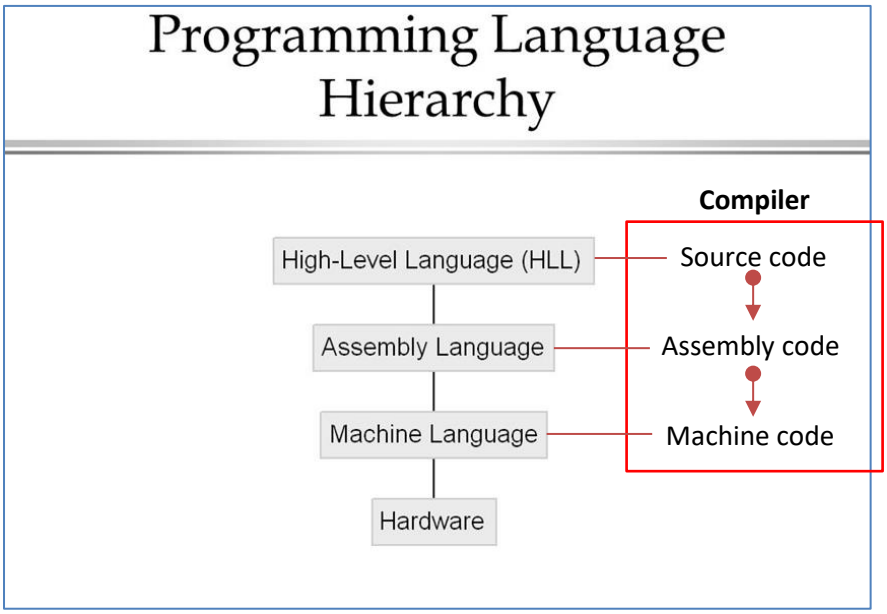



When you write code using a high-level language like C#, that code is referred to as **source code.** This source code runs through a compiler and in the end the compiler will generate **machine code** that can be processed by your CPU.

Compilers are usually built in to the 'programming software' that we mentioned earlier. We are going to start exploring programming software in the next lesson. Compilers run in the background of programming software, so we never have to worry about what it does how it converts our source code into machine code.

It is important to know that 1 line of C# code could result in hundreds or even thousands of lines of machine code (0s and 1s). Therefore, we would never write computer programs using low-level machine languages.

One last note about programming languages is that there is a hierarchy (ranking from top to bottom). As mentioned before, C# is a high-level language, and it sits at the top of the programming language hierarchy. Before high-level langue source code gets converted to machine language it first gets converted to **assembly language** (by a compiler as described above)**. Assembly** is a language used when dealing with hardware directly. Here is a diagram of the programming language hierarchy and the different conversions that take place when you **compile** the source code of a high-level language like C#:




**Note:** C# also uses the concepts of an **interpreter** which is like a compiler except that machine code does not get generated until runtime. We do not need to dive into the details of interpreters vs compilers. This course is more focused on using C#!

## III. THE C# PROGRAMMING LANGUAGE:

**C#** is a general-purpose, multi-paradigm programming language. C# encompasses static typing, strong typing, lexically scoped, imperative, declarative, functional, generic, object-oriented, and component-oriented programming disciplines (Wikipedia).

You probably have no idea what half those terms mean and that's ok!

Just know that C# is a language created by Microsoft and is widely used to program anything from websites to games to applications. It runs on something called the **.NET Framework,** which is a huge library of code that C# borrows from.

This is an introductory course and C# will be introduced very slowly. C# was chosen for this course because it is easy to use (compared other programming languages), and we can use it to make a game by the end of the course using the **Unity Game Engine** which is a highly popular C# gaming engine used by many game developers.

Before we start with C#, we will need to install some **programming software.** We will be using Microsoft's **free** programming software called **Microsoft Visual Studio Code.** The next lesson involves setting up our computer environment with MS Visual Studio Code so that we can get started with C#.

Happy coding!!!