1. How to block any website from the server

Using Windows Firewall

Using chrome locke

TP-Link Router

<https://www.youtube.com/watch?v=4SyqNW2s7YQ>

1. all types of inheritance and when to use each type

* Single inheritance

a derived class is created from a single base class.

* Multi-level inheritance

a derived class is created from another derived class.

* Multiple inheritance

a derived class is created from more than one base class. This inheritance is not supported by .NET Languages like C#, F#, etc., and Java Language.

* Multipath inheritance

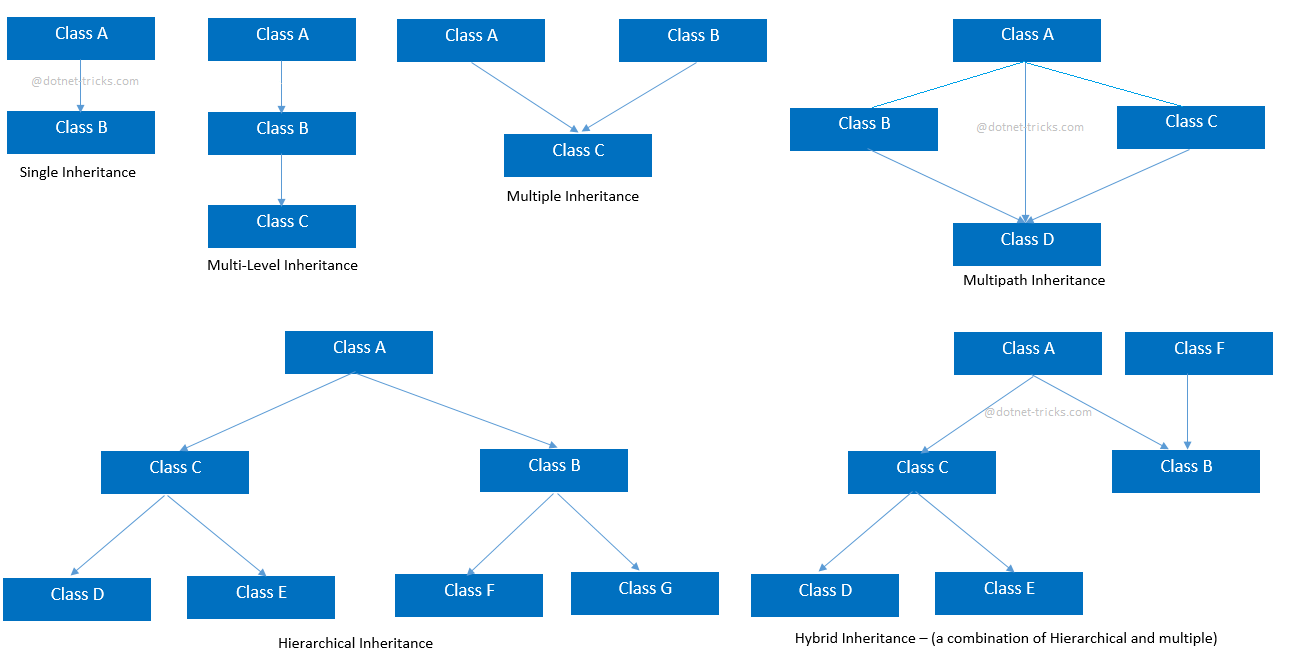
a derived class is created from other derived classes and the same base class of other derived classes. This inheritance is not supported by .NET Languages like C#, F#, etc.

* Hierarchical Inheritance

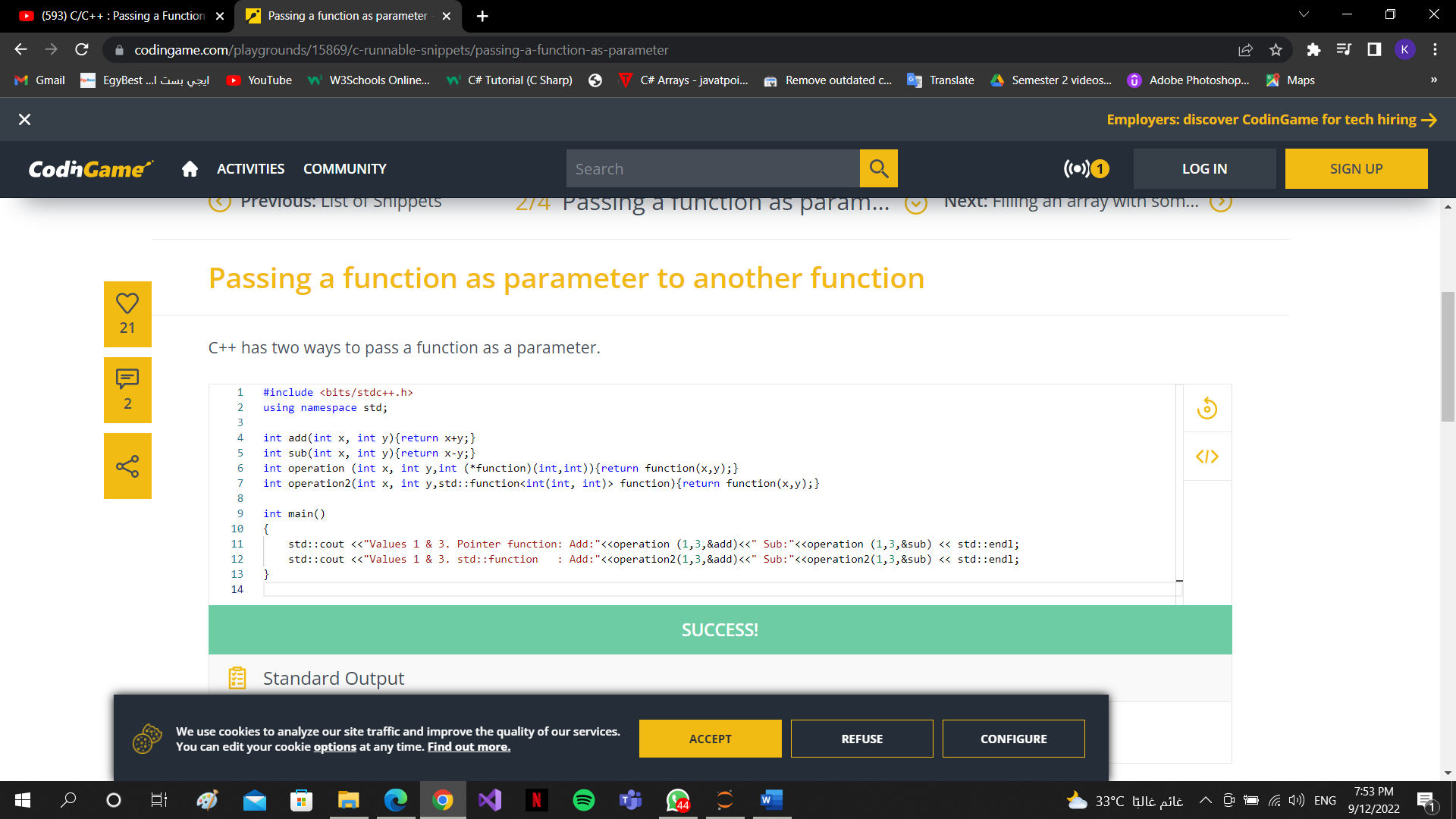
more than one derived class is created from a single base class and further child classes act as parent classes for more than one child class.

* Hybrid Inheritance

This is a combination of more than one inheritance. Hence, it may be a combination of Multilevel and Multiple inheritance or Hierarchical and Multilevel inheritance Hierarchical and Multipath inheritance, or Hierarchical, Multilevel and Multiple inheritances.



<https://www.dotnettricks.com/learn/oops/understanding-inheritance-and-different-types-of-inheritance>

1. how to pass function as a parameter to another function
2. Companies use von neumann architecture till now

the Von Neumann architecture is not used exclusively: almost any current "Von Neumann" machine except for very small microcontrollers (which are occasionally Harvard machines) features several important extensions to the original architecture, from DMA to MMUs.

Specialized coprocessors are very popular - most notably, GPUs.

Those work very well in conjunction to a Von Neumann machine.

The limitations of the Von Neumann architecture and the need for non-Von Neumann architectures for certain applications is well recognized in the scientific community, most importantly whenever "[general] artificial intelligence" of interest, see for example this very recent paper, which continues the line of work started by Carver Mead, who envisioned one of the most radical departures from "classical" architectures in 1989, in the form of analog, time-continuous, highly parallel "neuromorphic" chips.

1. Difference between framework and library and package

**framework**: Inversion of Control, The framework calls you, not the other way round. (This is known as the Hollywood Principle: "Don't call us, we'll call you.") The framework is in control. The flow of control and the flow of data is managed by the framework**.(collection of libraries and also has an architecture )**

**library**: collection of related functionality, is just a collection of related functionality. Nothing more, but also nothing less. The defining characteristic of a library is that you are in control, you call the library**. (collection of packages)**

**Package -** Collection of classes/files of similar functionality. **(Is collection of modules)**

**Module -**It is the smallest piece of software. It is set of methods/functions ready

to be used somewhere else.

**For example**, you might use an XML generator library when writing a web application using a web framework, and that XML library might have been provided by the framework or even be an integral part of it.

<https://stackoverflow.com/questions/4099975/difference-between-a-module-library-and-a-framework>

<https://www.youtube.com/watch?v=2j7dqE3qaLM>

1. how to write part of the code by framework and the other part not