Part 4 of home-assignment
Adva Kornblau

1. What is the difference between class and struct in C#?

In C#, the differences between a class and a struct concern the following categories:

Memory Model: A class is a reference type. Class instances are allocated on the heap, and accessed through a reference. The memory is managed by the garbage-collector. A struct is a value type, and its instances are usually allocated on the stack, and copied by value.

Inheritance: Class supports inheritance from another class, whereas struct cannot inherit from other structs or classes.

Polymorphism: Class supports polymorphism (virtual/override methods). Struct does not support polymorphism.

Constructors: A struct always has an implicit parameterless constructor that initializes all fields to default values (int -> 0 , bool -> false, reference fields -> null etc.).

In classes, the programmer can declare constructors, but if the programmer does not declare any constructors, a default constructor will be generated by the compiler.

Nullability: Class objects can be null. Struct instances cannot be null, unless they are declared as nullable.

2. What is a try-catch-finally block, and when should you use it?

The try-catch-finally block is used to handle exceptions at runtime, and avoid unhandled termination. Common exceptions are: DivideByZeroException, IndexOutOfRangeException, ArgumentNullException.

try: The block where you write the code that might throw an exception.

catch: Executes only if an exception is thrown. Handles the exception by logging diagnostic details, displaying a message, or rethrowing the exception.

finally: Always executes (whether an exception was thrown or not), typically used for resource cleanup (closing files, releasing database connections, etc.).

3. What is the difference between Select, Where, FirstOrDefault, and Any?

Those four LINQ methods are used to query and manipulate collections, but they serve different purposes:

Select: Projects each element of a collection, with a transform that the programmer applied.

Where: Filters a collection based on a condition , and returns only the matching elements in the collection.

FirstOrDefault: Returns the first element in the collection that matches a condition. If no element is found, returns the default value of the type.

Any: Checks whether the collection contains at least one element that matches a condition. return boolean value (true/false).

4. Explain the four principles of Object Oriented Programming: Encapsulation, Abstraction, Inheritance and Polymorphism.

Encapsulation: The principle of wrapping data and behaviors inside a class, and restricting access to internal details (with access permissions such as private, public, protected). limit access to private data members with getters and setters.

Abstraction: Conceal details of implementation from the user, and expose only the essential features. The user does not need to know how the function is implemented in order to use it.

Inheritance: A mechanism that allows a class to be derived from a base class. The derived class gains the base's class methods and functionality. The derived class can extend the base class by adding new data members and methods.

Polymorphism: The ability to use the same methods name through a common base class or interface, with a different behavior, depending on the context of the object (using virtual/override or interfaces).