Part 1: Constructors

A constructor is a special method inside a class that runs automatically whenever a new object is created. As stated in the Microsoft C# Documentation, constructors help set starting values so the object is ready to use immediately. This prevents the object from starting with empty or incorrect data, making the program safer and easier to maintain. If we ever need to change the default values, we only update the constructor instead of modifying multiple parts of the program.

Real-World Examples

- **Games:** When a new player or enemy is created, the constructor sets their default health, speed, or level (a common example shown in C# OOP tutorials).
- **Bank Applications:** As explained in W3Schools, when a bank account object is created, the constructor assigns the account number and starting balance.
- **Web or API Apps:** When connecting to an API, the constructor sets the base URL and API key so the setup does not need to be repeated.

Part 2: OOP Principle - Encapsulation

Encapsulation is an object-oriented programming principle that focuses on protecting data inside a class. According to standard OOP guides, we make variables private so they cannot be modified directly from outside. Instead, methods like SetName() or GetBalance() allow safe and controlled access. This improves security, reduces bugs, and ensures that changes to the data are valid.

Classes and Objects

A class works as a blueprint or design used to create objects. When we create an object from a class, it becomes an actual instance that can store data and perform operations.

Class: The design (e.g., a blueprint of a Dog).

Object: The real example created from that design (e.g., an actual dog).

Simple Examples

- **Banking System:** The balance is private. It can only be changed using deposit or withdraw methods, ensuring correct updates.
- **Student System:** Marks are stored privately and can only be updated through methods that prevent invalid values.

Source Used:

- **Microsoft C# Documentation**
  https://learn.microsoft.com/en-us/dotnet/csharp/programming-guide/classes-and-structs/constructors

- **W3Schools**
  https://www.w3schools.com/cs/cs_constructors.php