Prototype Design Pattern - Summary

The Prototype Design Pattern is a creational design pattern that allows us to create new objects by cloning existing ones.

This pattern is especially useful when object creation is costly or when many similar objects are needed.

Real-world Analogy:

Think of a racing game where you want to create a new car. Instead of building it from scratch, you can clone an existing car.

When to Use:

- When object creation is expensive or complex.
- When we need many objects with the same initial state.

How It Works:

1. Define a prototype interface with a clone method.
2. Implement this interface in a class (e.g., Car).
3. Use the clone method to create a copy of the object.

Code Execution Flow:

- Create an original object using the constructor.
- Use the clone method to create a copy of the original.
- Print both objects and compare their memory locations.

Observation:

Even though the cloned object has the same properties, it's a different object in memory (i.e., not the same reference).

This pattern simplifies the object creation process and is handy in games, document editors, or any domain requiring many similar objects.