**What is Polymorphism**?

means "many forms." In programming, this lets us use the same method name, like RecordEvent, on different kinds of objects and get different results depending on the object’s type. This is helpful because we can write code that works with a general type (like a Goal), but the program will still know which specific version of the method to run at runtime.

A big benefit of polymorphism is that it makes our code **more flexible and easier to maintain**. For example, if I later add a new type of goal, I don’t have to change the rest of my program. I just need to make sure the new goal type has its own version of the method.

An example from my own program is this line from GoalsManager.cs:

private void RecordEvent()  
{  
 ListGoalNames();  
 Console.Write("Which goal did you accomplish? ");  
 int target = Convert.ToInt32(Console.ReadLine()) - 1;  
  
 var goal = \_goals[target];  
  
 goal.RecordEvent();  
  
 \_score += goal.GetPoints();  
}

goal.RecordEvent();

This works whether goal is a SimpleGoal, EternalGoal, or ChecklistGoal. Each of those classes has its own RecordEvent() method, and the program automatically uses the right one. I don’t have to write separate code for each goal type when recording an event.

Polymorphism is powerful because it lets me **write cleaner code**, reuse logic, and work at a higher level without worrying about the details of each goal type. It really helped me stay organized while building my goal tracker.