**CSE 210: Programming with Classes**  
**W06 Assignment: Explain Polymorphism**

**What is polymorphism and why is it important?**

Polymorphism is an object-oriented programming principle that allows objects of different classes to be treated as objects of a common base class. It enables methods to behave differently based on the object that is calling them, even though the method name is the same. In simple terms, polymorphism allows one interface to be used for a variety of different data types.

A major benefit of polymorphism is that it makes code more flexible and scalable. It allows developers to write code that works with multiple types of objects without knowing their specific types in advance. This improves code reuse and makes it easier to manage large systems with related classes.

In my Week 6 project, I applied polymorphism in a goal-tracking program. I created a base class called Goal, and derived classes like SimpleGoal, EternalGoal, and ChecklistGoal. Each subclass had its own version of the RecordEvent and GetDetailsString methods. Even though I used the same method names, the behavior differed depending on the specific type of goal object.

Here’s a snippet that shows how polymorphism was used:

List<Goal> goals = new List<Goal>();  
goals.Add(new SimpleGoal("Read Scriptures", "Read daily", 100));  
goals.Add(new EternalGoal("Practice Piano", "Daily practice", 50));  
goals.Add(new ChecklistGoal("Attend Temple", "Go once a week", 75, 5, 500));  
  
foreach (Goal goal in goals)  
{  
 Console.WriteLine(goal.GetDetailsString());  
}

In this example, the GetDetailsString() method is called on each object in the list, but the version that gets executed depends on the actual type of the goal object. This is polymorphism in action.

In summary, polymorphism simplifies code, supports the open/closed principle (open for extension, closed for modification), and makes programs easier to extend and maintain.