# What is Polymorphism?

Polymorphism in object-oriented programming allows objects from different classes to be treated as if they are from the same base class. This enables methods to behave differently depending on the type of object they are used with. It simplifies the code and makes it easier to maintain since shared methods are stored in one place, avoiding repetition. For example, in the Eternal Quest program, the Goal class has a RecordEvent() method that behaves differently depending on whether the goal is a simple goal or an eternal goal. The base method is overridden by the methods defined in the derived classes.

Below is an example of how the SimpleGoal and EternalGoal classes override the RecordEvent() method to implement their own behaviours based on the goal type.

**Goal.cs**

public abstract void RecordEvent();

**SimpleGoal.cs**

public override void RecordEvent()

{

if (!\_isComplete)

{

\_isComplete = true;

Console.WriteLine($"Goal '{GetShortName()}' completed! You earned {GetPoints()} points.");

}

else

{

Console.WriteLine($"Goal '{GetShortName()}' is already completed.");

}

}

**EternalGoal.cs**

public override void RecordEvent()

{

    Console.WriteLine($"Goal '{GetShortName()}' recorded! You earned {GetPoints()} points.");

}