Polymorphism makes it possible to execute an action in different ways depending on the behavior that is required. For example, calculating the perimeter of a shape depends on the shape. The action, calculating the perimeter, is valid for all shapes, but the method of calculation will be different for a square vs. a rectangle vs. a circle. Polymorphism allows the needed technique to override the basic concept. The benefit of polymorphism is that, combined with the principle of Inheritance, it is possible to vary the behavior of a derived class within the context of the overarching behaviors of the base class.

In the goal tracking assignment this was shown in the following way:

The Goal class Record Event (marking goal complete) code was simply -

public abstract int RecordEvent();

The Simple Goal class recorded the event in this way –

public override int RecordEvent()

    {

        \_isComplete = true;

        return \_points;

    }

While the Checklist Goal class recorded it this way (a more complicated version) –

public override int RecordEvent()

    {

        \_amountCompleted++;

        if (!IsComplete())

        {

            return \_points;

        }

        else

        {

            Console.WriteLine($"Congratulations! You just earned {\_bonus} points!");

            return \_points + \_bonus;

        }

    }

The program was able to simply call the method RecordEvent() and, depending on the goal class, the correct method was run.