What is polymorphism and why is it important?

Polymorphism is a feature of object-oriented programming that gives elements the ability to perform differently based on the underlying context in which they are used. Along with inheritance and method overriding we can use polymorphism to implement common methods in child classes that can perform differently depending on the subclass.

An application of polymorphism could be calculation methods that take different types of shapes as parameters. For example, we can have a CalculateArea() method in a Shape superclass, and for each of its subclasses that represent a different kind of shape, we can implement a different way to calculate the shape area.

Here is another example of an application from the code of my Goal Tracker program:

**public abstract class Goal**

{ public abstract void NewEvent(); }

*// The superclass Goal implements an abstract method called NewEvent() that handles the recording of the completion of a Goal.*

**public class SimpleGoal : Goal**

{

public override void NewEvent()

{

if (!\_isCompleted)

{

\_isCompleted = true;

\_timesCompleted = 1;

}

else

{

Console.WriteLine("(!) This goal is already completed!");

}

}

*// The subclass SimpleGoal (and all of the Goal subclasses) overrides the method and implements its own syntax for handling the NewEvent recording.*

**public class EternalGoal : Goal**

{

public override void NewEvent()

{

\_timesCompleted += 1;

}

}

public class ChecklistGoal : Goal

{

public override void NewEvent()

{

if (!\_isCompleted)

{

\_timesCompleted += 1;

if (\_timesCompleted == \_repetitionsToBonus)

{

\_isCompleted = true;

}

}

else

{

Console.WriteLine("(!) This goal is already completed!");

}

}

**\_userGoals[goalToCompleteIndex].NewEvent();**

*// Finally we are accessing a Goal inside the “\_userGoals” List, which could be of any of the Goal subclasses types, and for the resulting Goal we are calling the “NewEvent()” method, which will implement different syntaxes according to the type of Goal stored in the given index “goalToCompleteIndex”*