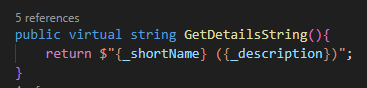
**Polymorphism**

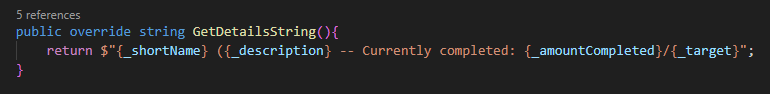
Polymorphism in programming is when a line of code behaves in different ways depending on the context. This technique is possible because of method overriding, which is when a child class overrides a method with the same name as the parent class.

Sometimes a child class must have a method from the parent class, but does not necessarily have to behave the same way. Instead of creating another method for the child class with a different name, we can just override the parent class’s method, and add new behavior. This simplifies the code, makes it more readable, and easier to use in a runtime.

For example, in this week’s Prove assignment, there is a parent class Goal, and two of the child classes, EternalGoal and ChecklistGoal. The Goal class has a method called GetDetailsString() which retrieves the formatted details from that goal:



The EternalGoal Class has no problem with this method staying like this, so it doesn’t have to be changed. Once referenced by the EternalGoal class, it wil behave just like the Goal class stated. On the other hand, the ChecklistGoal class can’t use this behavior, because it must also display the \_amountCompleted and the \_target variables. To solve that, we must just override the parent’s class, and add new behavior:



Now, once we reference the same method on those 2 different classes, they will behave differently:

