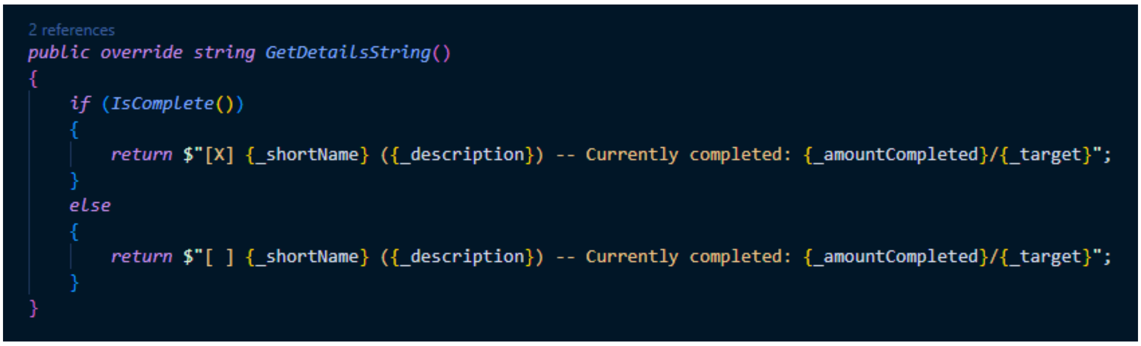
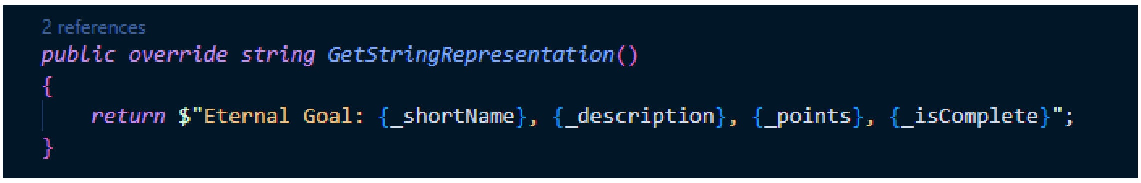
Polymorphism is basically having methods with the same name work in different ways depending on what object you're dealing with. In my project, this shows up because each type of goal (like ChecklistGoal or EternalGoal) takes the basic methods from the Goal class and redefines them to do what’s needed for that specific goal.

For example, look at the ChecklistGoal's GetDetailsString() method. It checks if the goal is complete and prints out the progress:



```
2 references
public override string GetDetailsString()
{
    if (IsComplete())
    {
        return $"[X] {_shortName} ({_description}) -- Currently completed: {_amountCompleted}/{_target}";
    }
    else
    {
        return $"[ ] {_shortName} ({_description}) -- Currently completed: {_amountCompleted}/{_target}";
    }
}
```

Now, for an EternalGoal, the GetStringRepresentation() method is used to show that it’s an eternal goal:



```
2 references
public override string GetStringRepresentation()
{
    return $"Eternal Goal: {_shortName}, {_description}, {_points}, {_isComplete}";
}
```

Because of polymorphism, the GoalManager can just call RecordEvent() on any goal from a list without worrying about which kind it is. This makes the code simpler and easier to build on later.