What is polymorphism and why is it important?

Author : Diogo Rangel Dos Santos

Your response must:

- **Explain the meaning of Polymorphism.**

Polymorphism is a principle of object-oriented programming (OOP) that allows objects of different derived classes to be treated as objects of a common base class. The word "polymorphism" comes from Greek, meaning "many forms." In programming, this means that a single interface or method can represent different underlying data types or behaviors.

In C#, polymorphism is typically implemented using method overriding (with virtual and override) and interfaces. This allows for flexible and scalable code where one interface can be used with different types of objects.

- **Highlight a benefit of Polymorphism.**

One major benefit of polymorphism is that it allows for extensibility and code reuse. You can write code that works on the base class and it will automatically work with any derived classes. This simplifies your logic and makes your code easier to maintain, modify, and expand.

- **Provide an application of Polymorphism.**

In the Eternal Quest program, polymorphism is used when different types of goals (SimpleGoal, EternalGoal, ChecklistGoal) all derive from a base class Goal. Despite having different logic for completion and scoring, they can be used interchangeably when stored in a list or when invoking a shared method like RecordEvent() or GetDetailsString().

- **Use a code example of Polymorphism from the program you wrote. (You should copy and paste a few lines of code that demonstrate the use of the principle.)**

```csharp
44    class SimpleGoal : Goal
45    {
46        private bool _isComplete;
47
48        public SimpleGoal(string name, string description, int points, bool isComplete = false)
49            : base(name, description, points)
50        {
51            _isComplete = isComplete;
52        }
53
54        public override int RecordEvent()
55        {
56            if (!_isComplete)
57            {
58                _isComplete = true;
59                return _points;
60            }
61            return 0;
62        }
63
64        public override string GetDetails()
65        {
66            return ($"[{(_isComplete ? "X" : " ")}] {_name} ({_description})");
67        }
68
69        public override bool IsComplete() => _isComplete;
70
71        public override string Serialize()
72        {
```