Explaining polymorphism:

Polymorphism refers to the programming concept that enables objects from various classes to exhibit different behaviors while being treated as instances of a common base class. A key element of polymorphism is method overriding, which allows a child class to redefine or customize the behavior inherited from its parent class. This principle is considered the pinnacle of object-oriented programming, as it integrates abstraction, encapsulation, and inheritance to function effectively.

The primary benefit of polymorphism is that it enhances code reusability and maintainability. It simplifies the process of extending or updating programs without altering the existing codebase.

An example of polymorphism in action can be seen in the mindfulness activity, where different types of objects share a common behavior. This demonstrates how polymorphism allows diverse objects to operate within a unified framework while maintaining their unique implementations. This is an example of polymorphism below:-

*public abstract class MindfulnessActivity*

*{*

*public abstract void RunActivity();*

*}*

*public class BreathingActivity : MindfulnessActivity*

*{*

*public override void RunActivity()*

*{*

*Console.WriteLine("Inhale... Exhale...");*

*}*

*}*

*public class ReflectionActivity : MindfulnessActivity*

*{*

*public override void RunActivity()*

*{*

*Console.WriteLine("Think about a moment of personal growth...");*

*}*

*}*

*// Using polymorphism*

*public void StartActivity(MindfulnessActivity activity)*

*{*

*activity.RunActivity(); }*