

# What is Polymorphism?

In this lesson, the concept of Polymorphism will be explained which is an important concept in OOP.

## We'll cover the following

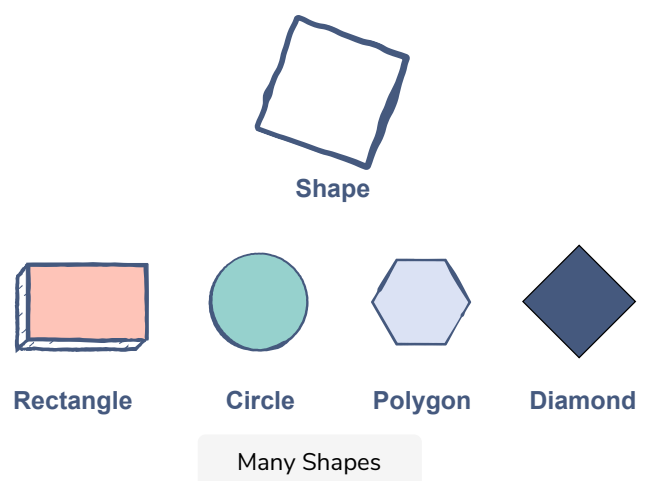
- Definition
- A Brief Introduction
  - Make Things Simpler with Polymorphism
  - What does Polymorphism Achieve?

## Definition #

The word **Polymorphism** is a combination of two Greek words, **Poly** meaning *many* and **Morph** meaning *forms*.

In programming, *polymorphism* refers to the same object exhibiting different forms and behaviors.

For example, take the Shape Class. The exact shape you choose can be anything. It can be a rectangle, a circle, a polygon or a diamond. So, these are all shapes, but their properties are different. This is called **Polymorphism**.



## A Brief Introduction #

Assume there is a parent class named **Shape** from which the child classes

`Rectangle`, `Circle`, `Polygon`, and `Diamond` are derived.

Suppose your application might need methods to calculate the area of each specific shape. Of course, the area for each shape is calculated differently. So, you can't have a single implementation. You could throw in separate methods in each class called, for instance, `getSquareArea()`, `getCircleArea()` etc. But this makes it harder to remember each method's name. Wouldn't it be nice if all specific area calculation methods could be called `getArea()`?

## Make Things Simpler with Polymorphism #

You would only have to remember one method name and when you call that method, the method specific to that object would be called. This can be achieved in object-oriented programming using polymorphism. The base class declares a function `getArea()` without providing an implementation. Each derived class inherits the function declaration and can provide its own implementation

Consider that the Shape class has a method called `getArea()`, which is inherited by all subclasses mentioned. With polymorphism, each subclass may have its own way of implementing the method. So, for example, when the `getArea()` method is called on an object of the `Rectangle` class, the method will respond by displaying the area of the rectangle. On the other hand, when the same method is called on an object of the `Circle` class, the circle's area might be calculated and displayed on the screen.

## What does Polymorphism Achieve? #

In effect, polymorphism cuts down the work of the developer. When the time comes to create more specific subclasses with certain unique attributes and behaviors, the developer can alter the code in the particular portions where the responses differ. All other pieces of the code can be left untouched.

---

So far, we've learned what polymorphism is. In the next lesson, we will learn how to implement polymorphism in OOP.