

Method Overriding

In this lesson, you'll be learning about what method overriding is and how to achieve it in Python.

We'll cover the following



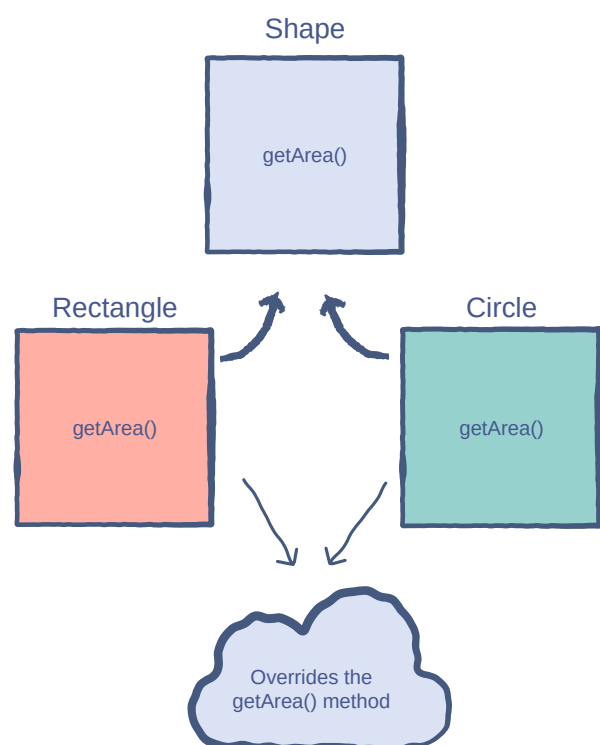
- A Brief Introduction
- Advantages and Key Features of Method Overriding

A Brief Introduction

Method overriding is the process of redefining a parent class's method in a subclass.

In other words, if a subclass provides a specific implementation of a method that had already been defined in one of its parent classes, it is known as **method overriding**.

In the previous (<https://www.educative.io/collection/page/10370001/6201068373409792/5676337998069760>) example, the Rectangle and Circle classes were overriding the `getArea()` method from the Shape class.



In this case:



- The method in the parent class is called **overridden method**.
- The methods in the child classes are called **overriding methods**.


We have already seen the implementation of the `getArea()` method in the previous lesson (<https://www.educative.io/collection/page/10370001/6201068373409792/5676337998069760>), which depicts the concept of overriding. The *highlighted* portions show where method overriding is happening.

Let's have a look!

```
1 class Shape:
2     def __init__(self): # initializing sides
3         self.sides = 0
4
5     def getArea(self):
6         pass
7
8
9 class Rectangle(Shape): # derived form Shape
10    # initializer
11    def __init__(self, width=0, height=0):
12        self.width = width
13        self.height = height
14        self.sides = 4
15
16    # method to calculate Area
17    def getArea(self):
18        return (self.width * self.height)
19
20
21 class Circle(Shape): # derived form Shape class
22    # initializer
23    def __init__(self, radius=0):
24        self.radius = radius
25
26    # method to calculate Area
```



```
27 def getArea(self):
28     return (self.radius * self.radius * 3)
29
30
31 shapes = [Rectangle(6, 10), Circle(7)]
```



Advantages and Key Features of Method Overriding

- The derived classes can give their own specific implementations to inherited methods without modifying the parent class methods.
- For any method, a child class can use the implementation in the parent class or make its own implementation.
- Method Overriding needs inheritance and there should be at least one derived class to implement it.
- The method in the derived classes usually have a different implementation from one another.

Now that we are familiar with the concept of method overriding, let's understand the operator overloading in the next lesson.

[← Back](#)[Polymorphism Using Inheritance](#)[Next →](#)[Operator Overloading](#) Completed[Report an Issue](#)[Ask a Question](#)https://discuss.educative.io/tag/method-overriding__polymorphism__learn-object-oriented-programming-in-python

