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

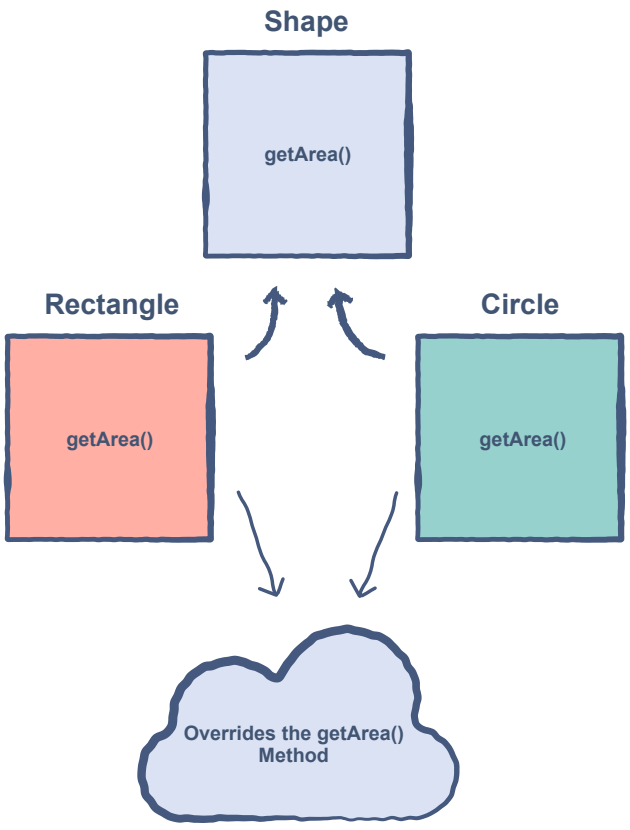
> *Method overriding* is the process of redefining a parent class's method in a subclass.

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 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, which depicts the concept of overriding. The *highlighted* portions show where method overriding is happening.

Let's have a look!

```
class Shape:
    def __init__(self):  # initializing sides of all shapes to 0
        self.sides = 0

    def getArea(self):
        pass


class Rectangle(Shape):  # derived form Shape class
    # initializer
    def __init__(self, width=0, height=0):
        self.width = width
        self.height = height
        self.sides = 4

    # method to calculate Area
    def getArea(self):
        return (self.width * self.height)


class Circle(Shape):  # derived form Shape class
    # initializer
    def __init__(self, radius=0):
        self.radius = radius

    # method to calculate Area
    def getArea(self):
        return (self.radius * self.radius * 3.142)


shapes = [Rectangle(6, 10), Circle(7)]
print("Area of rectangle is:", str(shapes[0].getArea()))
print("Area of circle is:", str(shapes[1].getArea()))
```

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.