

## Methods in Behavioral Research

Interface methods are the methods that have the purpose of providing an interface for an object with the external environment, for example, other objects' methods, data input from a user, data from another object, or anything that's not inside that very same object.

One of the principles of object-oriented design is encapsulation. Encapsulation is the technique of building an object to be a capsule containing all of its data and methods inside itself. However, an object that is isolated from everything is just plain useless, it should be a part of a bigger system. That's when interface methods come in: they provide the minimum necessary interface for that object to get external input and provide output so it can indeed be a part of a bigger system while also being a smaller system on itself.

For example, it is a very good object-oriented practice to define a getter and a setter method for an attribute, most notably to security and code integrity reasons. A getter gets the variable from the user input, and the setter assigns the variable as it will be used throughout the class. Getters and setters are methods which provide internal attributes access to external agents in an indirect way. Let's see how they are implemented:

```
Getter & Setter Interface Methods in Python: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
16 17 18 19 20 class Square ( object ): def __init__ ( self , sideLength ): self
. sideLength = sideLength # Getter: def getSideLength ( self ): return self .
sideLength # Setter: def setSideLength ( self , sideLength ): self . sideLength
= sideLength square = Square ( 10.0 ) print square . getSideLength () # prints
10.0 to the screen square . setSideLength ( 3.0 ) print square . getSideLength
() # prints 3.0 to the screen
```

This code snippet defines a Square class which will have instances with a sideLength attribute accessible through the getter method getSideLength which will provide that attribute's value and through the setter method setSideLength which will provide a way to alter that attribute's value.

Then, an instance of the Square class is created with an initial value for sideLength equaling 10.0 and this instance gets assigned to the square variable. The value of the sideLength variable is then printed on the screen, then the value of sideLength is updated and finally, the updated value of the sideLength variable is printed to the screen.

## Reference

[G is for Genes: The Impact of Genetics on Education and Achievement](#)

[An Introduction to Theories of Learning](#)