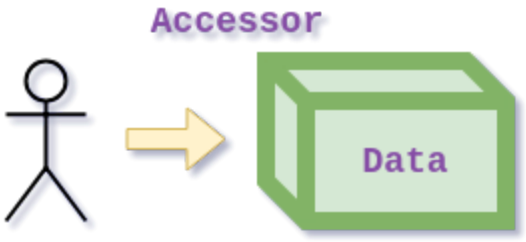
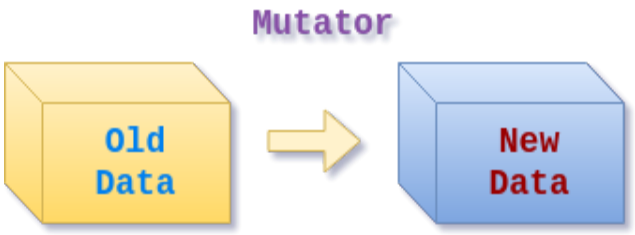
Accessor/Mutator



An accessor function of a class is a function that returns a value of a private member variable. These functions are listed under the **public** section of the class and are usually an inline function meaning the definition is on the same line as the declaration. They are usually named **getVAR()** where **VAR** is the variable name to return. An example of this is below:

```
class Integers
{
    public:
        //Constructors and Member Functions go here
        int getX() { return x; } //Accessor Function
        int getY() { return y; } //Accessor Function
    private:
        int x;
        int y;
}; //Accessor functions can be defined in class implementation as well
```



A mutator function of a class is a function that changes a value of a private member variable. These functions are listed under the **public** section of the class. They are usually named **setVAR(Formal Parameter)** where **VAR** is the variable name to return, or **set(Formal Parameters)** if more than one variable is to be set. Don't forget you are able to overload functions. An example is shown below:

```
class Integers
{
    public:
        //Constructors and Member Functions go here
        void set(int xValue, int yValue) {x = xValue; y = yValue;}
        void set(int xValue, int yValue, int zValue); //Overloaded
        void setX(int xValue) { x = xValue; } //Function can be inline
    private:
        int x;
        int y;
        int z;
}; //Mutator functions can be defined in class implementation as well
```

Accessors may be known as **getters** and mutators may be known as **setters**.