Accessors and Mutators

Lesson 2.2

Learning Outcomes

LO 2.2.1	**Implement** necessary accessors on class definitions
LO 2.2.2	**Implement** necessary mutators on class definitions
LO 2.2.3	**Facilitate** object encapsulation through proper use of access modifiers, accessors, and mutators

Accessors

Accessors are methods designed to retrieve information.

They should have a return value therefore, should **not** have a **void** return type. The return value is closely related or the data type of the information/attribute that you are trying to access.

Accessors

Examples:

```
private String name;
public String getName()
{
    return this.name;
}
```

```
private int x;
public int getX()
{
    return this.x;
}
```

Mutators

Mutators are methods designed to set or customize information.

They usually have a **void** return type, but a return value may be returned representing a feedback. The parameters of the mutators should be able to contain the information you are trying to set.

Mutators

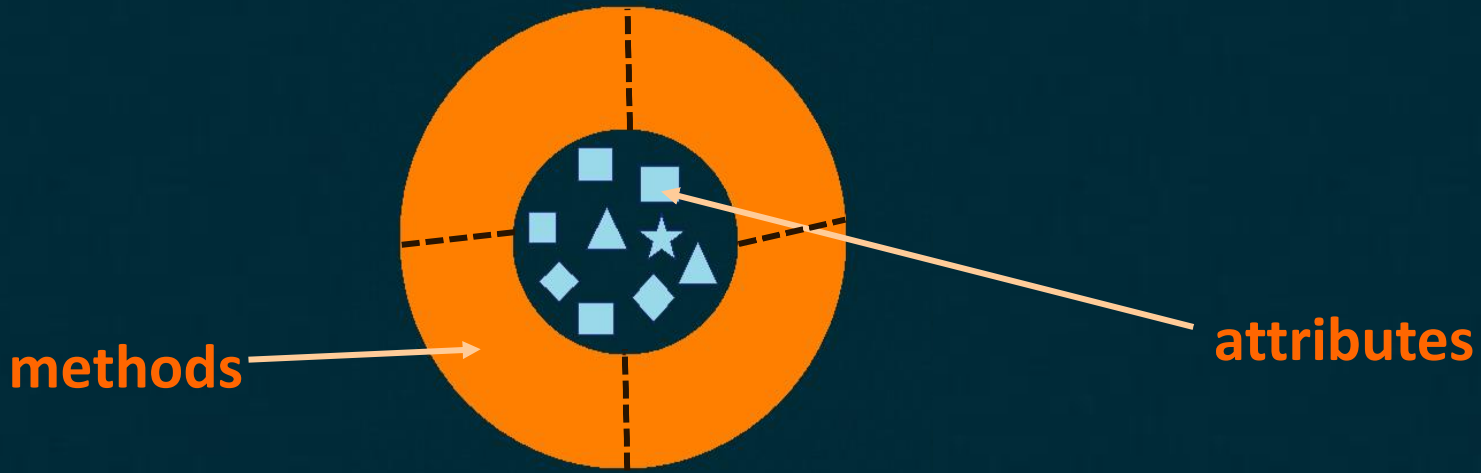
Examples:

```
private int x;
public boolean setX(int x)
{
    if(x >= 0)
        this.x = x;

    return x >= 0;
}
```

```
private String name;
public void setName(String
                      name)
{
    this.name = name;
}
```

Encapsulation



Encapsulation is a language construct that facilitates the bundling of data with the methods operating on that data.

"Encapsulation ***facilitates****; but does not guarantee, information hiding."* – Stephan Roth

LO 2.2.1 Implement necessary accessors on class definitions

Implement necessary accessors on class Quiz.

```
class Quiz
{
    Student owner;
    char[] answers;
    char[] correct_answers;
    int score;
}
```

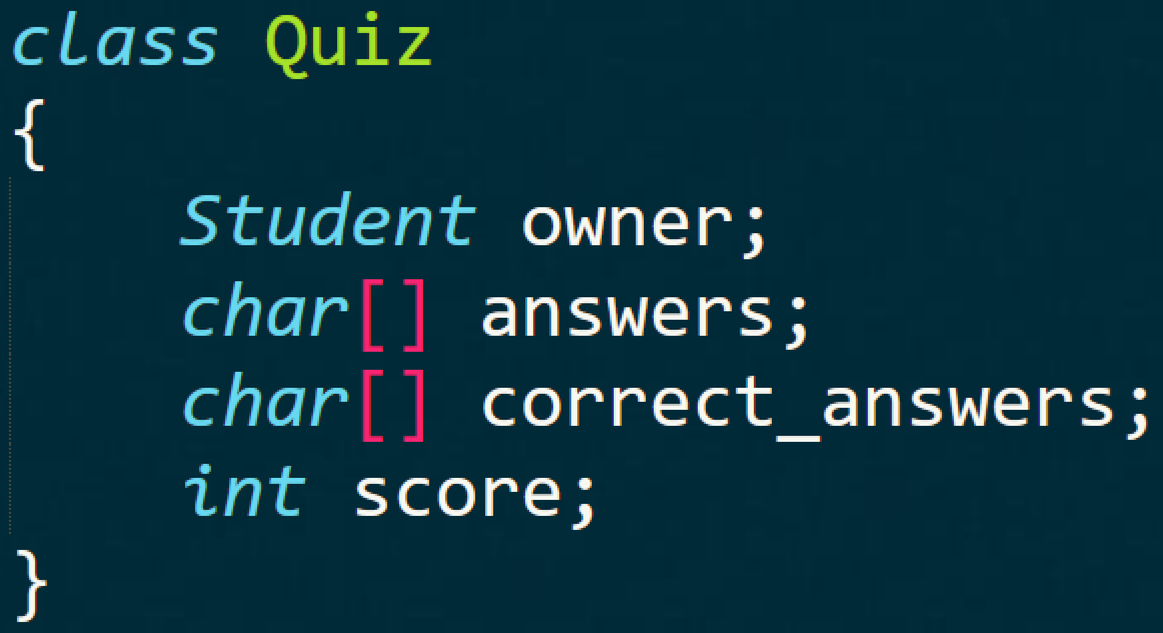
LO 2.2.2 Implement necessary mutators on class definitions

Implement necessary mutators on class Quiz.

```
class Quiz
{
    Student owner;
    char[] answers;
    char[] correct_answers;
    int score;
}
```

LO 2.2.3 Facilitate object encapsulation through proper use of access modifiers, accessors, and mutators

Finalize encapsulating class Quiz.



```
class Quiz
{
    Student owner;
    char[] answers;
    char[] correct_answers;
    int score;
}
```