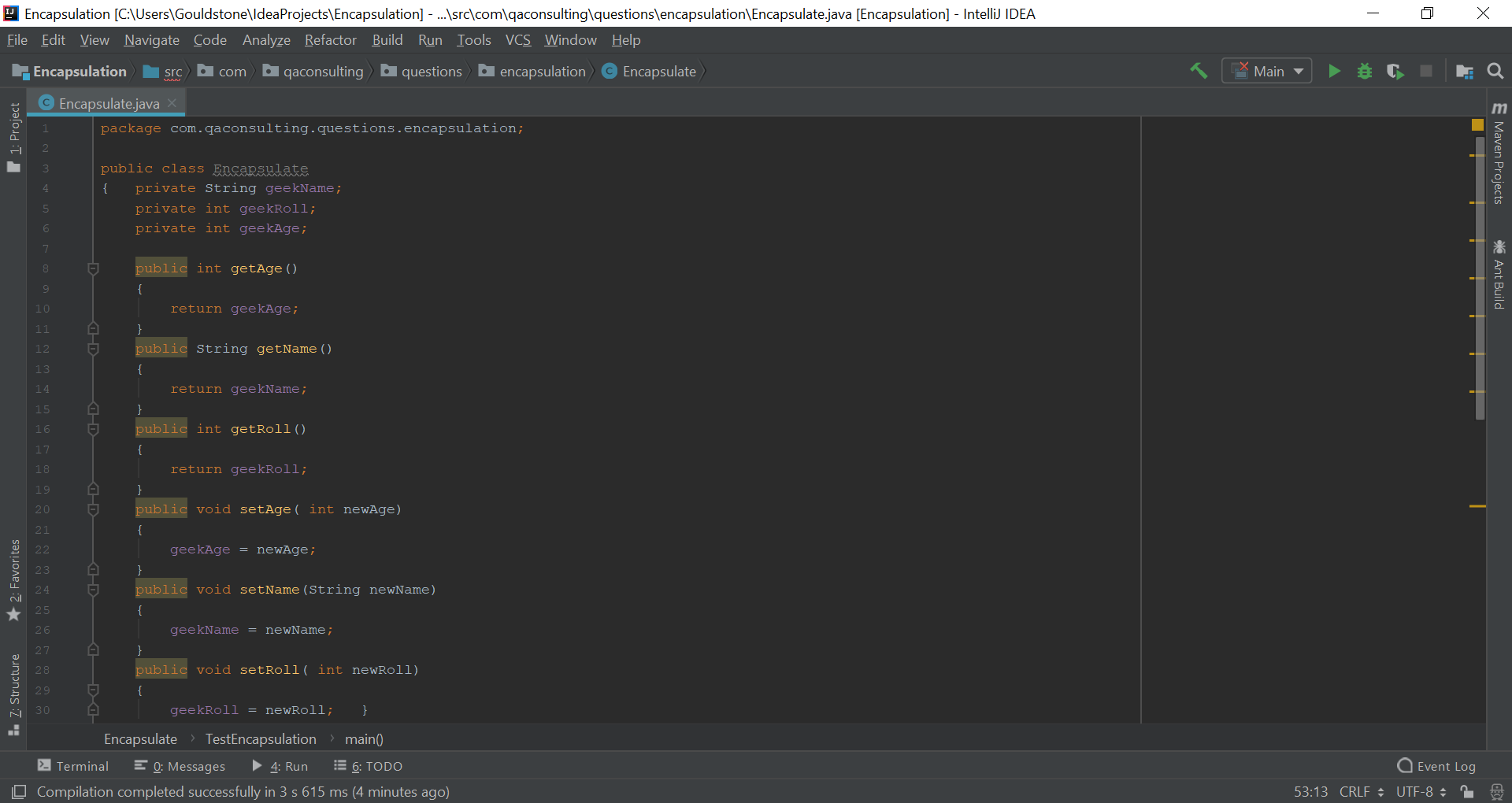
**QA Consulting Pre-Training Questions**

Exercise 1.

**Encapsulation** – This is the process of wrapping data and code together into a singular unit. A real-life example of this is in seen in the pharmaceuticals field when several different medicines are mixed within the same capsule. Creating a fully encapsulated class in Java is done by making all the data members of the class private and using setter and getter methods to retrieve the data. The Java Bean class is a perfect example of a fully encapsulated class, this class encapsulates many objects into a single object known as the bean. Some advantages of using the encapsulation process is that it allows the control of data, an example of this is if you wanted to set the value of id>100 only it will allow you to write the logic inside the setter method. An encapsulated class is easier to test so this would make it much more efficient for the likes of unit testing.

An example of this class is shown below;

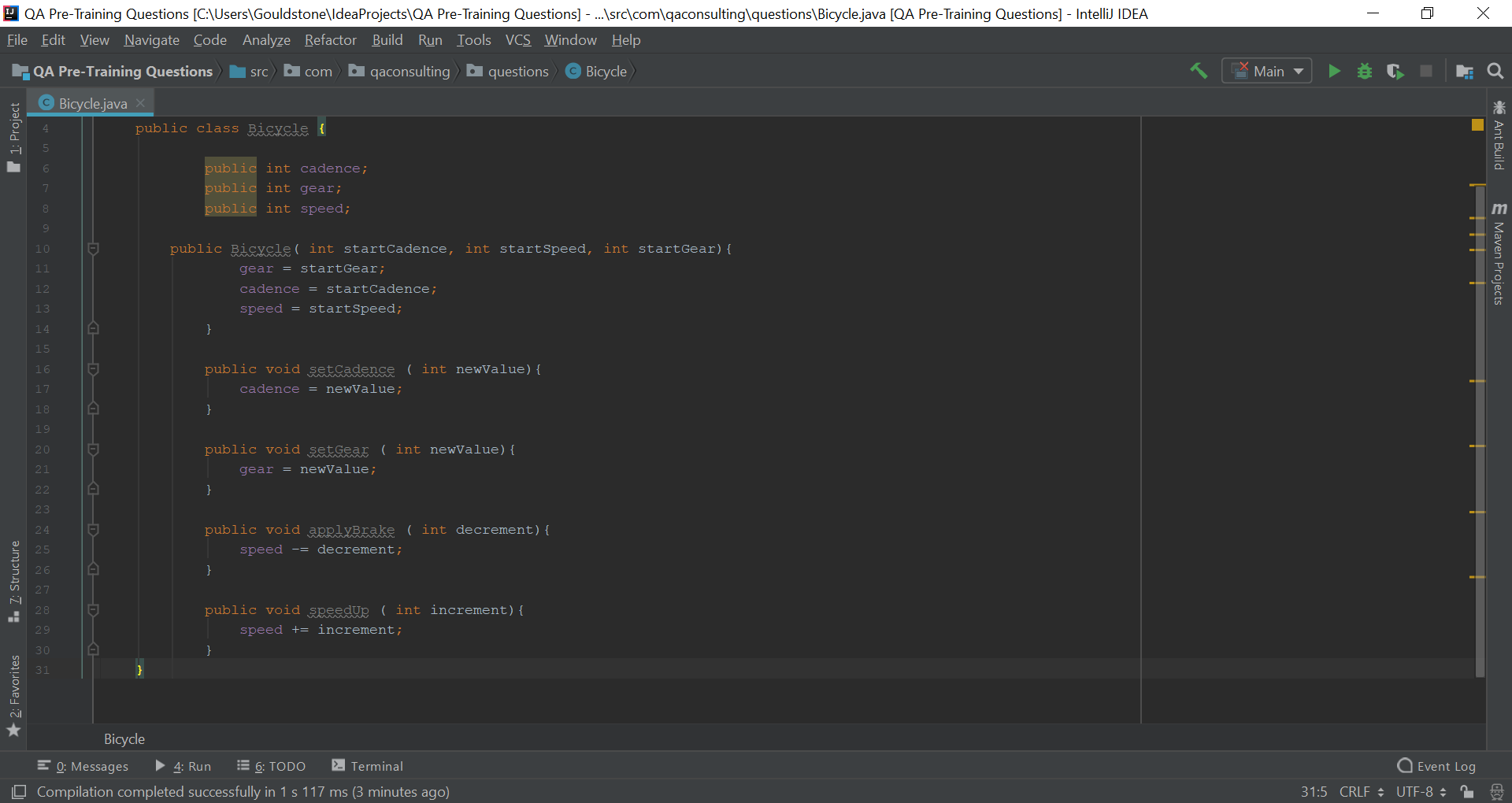


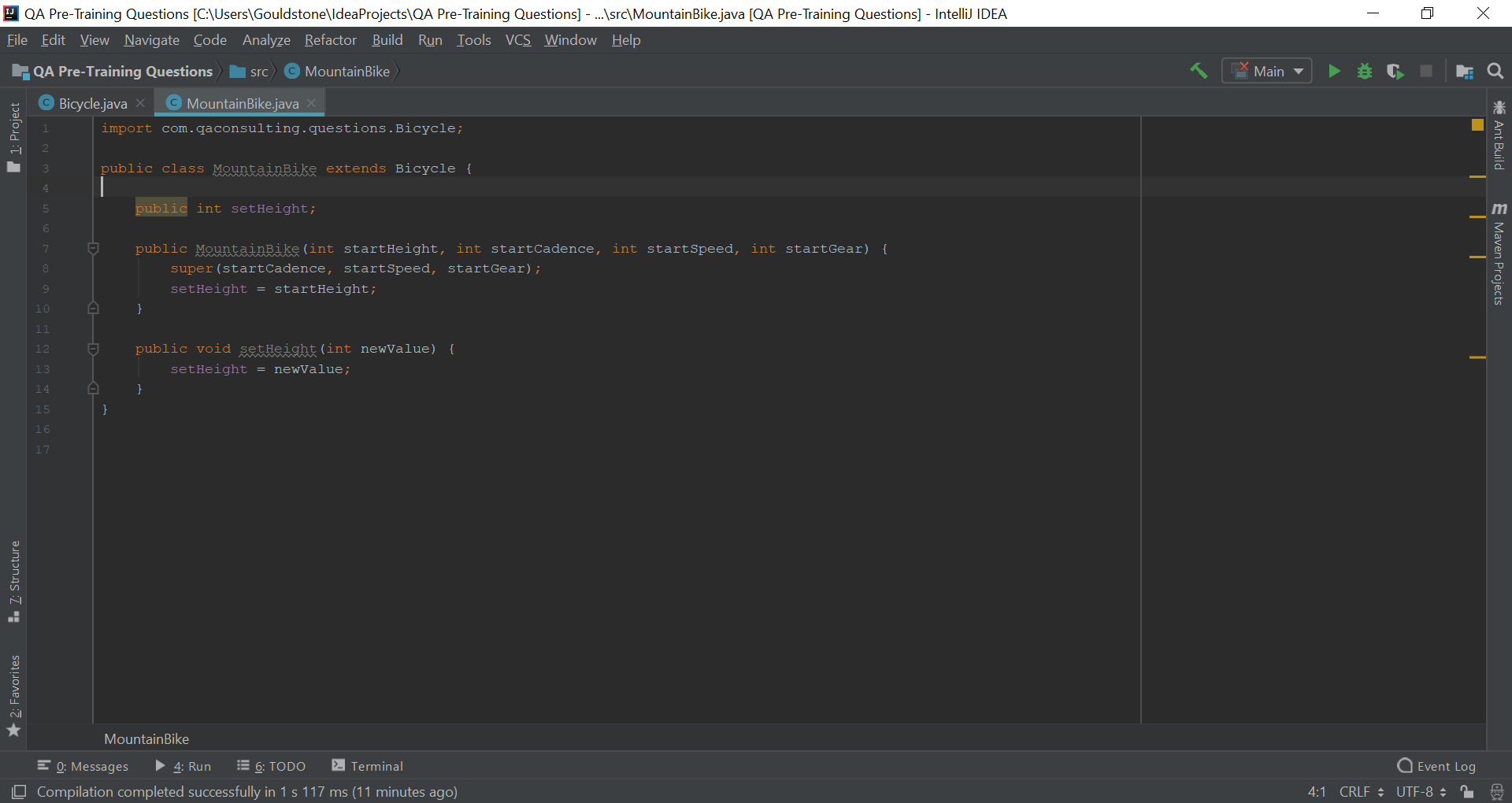
A screenshot of a computer screen

Description generated with very high confidence

**Inheritance** - This is the process by which one class can inherit the likes of fields and methods from another class. The class that inherits the fields and methods is called the subclass and the class from which the subclass inherits its properties and methods from is called the superclass. This process offers the ability to derive new classes from already existing classes, if the code in the existing class is needed for the new class. This enables you to reuse the fields and methods of the superclass without having to write them yourself.

An example of inheritance is shown below;





**Polymorphism** – This is known as the ability of an object to take on many forms. This is most commonly used in Java for reference of a parent class being used to refer to a child class.

An example of this is shown below;

A screenshot of a computer screen

Description generated with very high confidence

A screenshot of a computer screen

Description generated with very high confidence

**Abstraction -** Data abstraction is the process by which the identification of only the required characteristics of an object whilst ignoring the irrelevant details. In Java this is achieved using interfaces and abstract classes, but interfaces allow for 100% abstraction.

An example of abstraction is shown below;

A screenshot of a computer screen

Description generated with very high confidence

A screenshot of a computer screen

Description generated with very high confidence