Oracle offers excellent information on defining classes and instantiating objects through the Java Tutorials documents. In the section about declaring classes it explains that classes are defined within the class body; the field, constructor, and the method declarations are a means for an objects’ life cycle. The author explains this very well; constructors are for initializing new objects, declarations for the fields that provide the state of the class and its objects, and methods to implement the behavior of the class and its objects (Oracle, 2012). This is a minimal definition but it encases the required components of the class definition. Public and private are also class definitions, but there is much more that can define a class like the parent of the class and which interfaces the class implements. Let me also mention that a class name must be capitalized and the class body is to be surrounded by curly braces.

As far as object instantiation goes, Oracle had much to offer on this as well of course. Objects are created from a class and a class is a blueprint for the object. Once a variable declaration is made an object can be associated with it and when the new keyword (Java operator) is used it instantiates or creates the object. The object can then be initialized when the new operator is followed by a call to the constructor (Oracle, 2012).

<http://docs.oracle.com/javase/tutorial/java/javaOO/classdecl.html>

<http://docs.oracle.com/javase/tutorial/java/javaOO/objectcreation.html>