

## Notes and Definitions for OOP

- In Java, there are **primitive** types and **reference** types. The primitive types always start with a lowercase letter (int, double, boolean, etc). Reference types always start with an uppercase letter (although array types are also reference types: int[], double[], etc).
- OOP lets us define our own data types. These data types let us combine simple data types into more complex ones. These data types have **state** (the variables that are part of the data type) and **behavior** (the **methods** [functions] that are a part of the data type).
- In OOP, these new data types we can create are called **classes**. Classes are always reference types.
- The values of a new class data type are stored in **objects**.
- Creating a new object is called **instantiating** that object, which creates an **instance** of the corresponding class. A **constructor** is a special instance method that is used to construct an object.
  - It is easy to mix up what a class is versus what an object is. The class is the data type, whereas the object is the specific value. There can be many simultaneous objects in a program all corresponding to the same class.
- The most important difference in Java between primitive types and reference types (classes) is that you can invoke (call) the instance methods of a class by using a period to specify the object on which you want to call the method:

*object.methodName(param1, param2, etc...)*