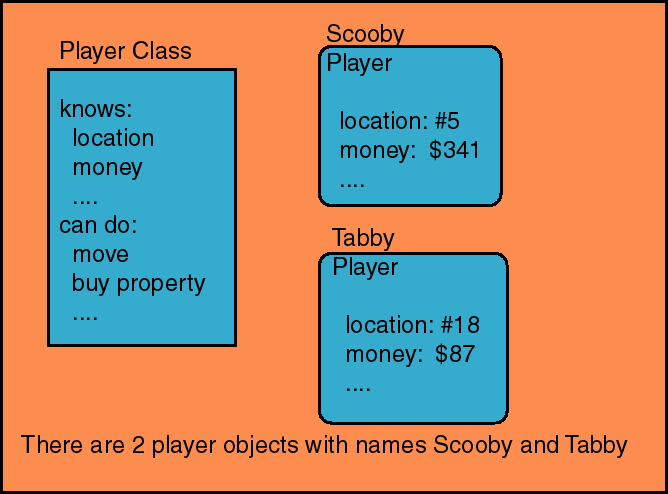
**What is the difference between a Class and an Object in Java?**

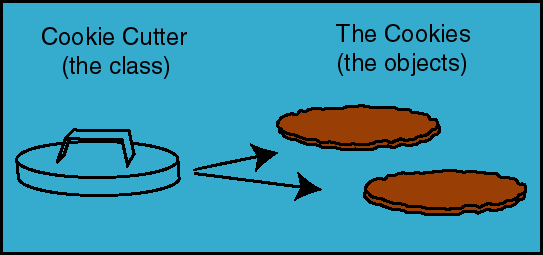
This article is solely for all beginner programmers, who are learning object oriented programming language e.g. Java, C++ or C# and aspire to do well on any programming interview. *Difference between class and object* is one of the most common question, you would like to ask a fresher coming out from college or training institute, but you would be surprised how many beginner Java programmers struggle with this question. Class and Object are two pillars of Object Oriented Programming (OOPS) and a good understanding is must, but when you ask this question apart from theoretical and bookish answer that *"class is a blueprint and objects are actual things created out of those blueprints"*, you would hardly get anything substantial. Though that answer is correct and works perfectly, it doesn't differentiate between a programmer, who has just mugged the answer, or the one who truly understand the difference between class and object. So, if I receive that answer, I usually ask them to create a class and explain how objects are created in program, let's say to represent an Employee, Student or simply a Car. If programmer truly understands [what is a class](http://java67.blogspot.sg/2012/08/what-is-class-file-in-java-how-to-create-class.html) and [what is an object](http://java67.blogspot.sg/2012/12/object-in-java-and-OOPS-example-tutorial.html), it will do that in no time, but if he has just mugged up the answer, then he will be totally confused. Point is to understand the difference between class and object, not to mug up for an interview, Why? because if you know your class and object, it would be lot easy for you to work with an object oriented programming language like Java or C#. To give you some more example of class and object, if Car is a class than Merc, Audi and BMW are objects. If Television is class than Sony Bravia, Samsung Smart tv are its object. If Smartphone is a class then  iPhone, Samsung Galaxy and Nokia Lumia are their object. In an object oriented application, generally nouns are represented using class, for example in finance domain Order, Trade, Instruments are classes. In E-commerce domain Payment, Order, Products are some example of classes.

## Class vs Object in Java

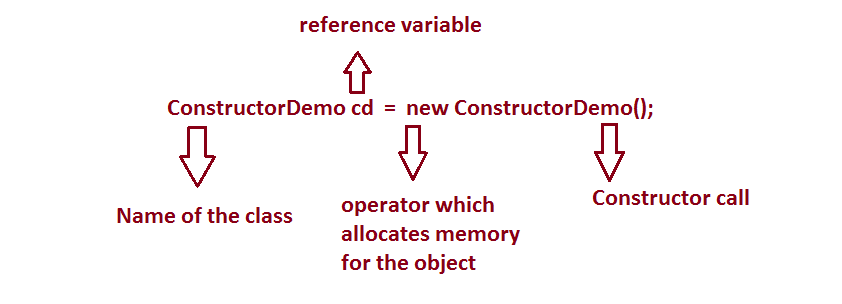
Some difference between class and object, which is totally based upon practical experience :  
  
1) A class is what you create while coding, but object is created at runtime by your execution environment e.g. JVM. Though you write code, which is required to create object during coding e.g. new Student(), object is not created at that time. They are only created when you run your program, and when runtime executes that line. Usually [constructor](http://java67.blogspot.sg/2012/11/java-enum-example-with-constructor.html) of a class is called when an object is created in Java, but yes there are some anomalies as well e.g. Serialization.  
  
2) Most important difference between class and object is that an Object usually has state (though stateless object is also possible). This is used to differentiate with another object. For example, If you have a class to represent Student, then John and Mohan are two object of that class, which has different name, an attribute which differentiate them. A picture is worth more than 1000 words, and *difference between class and object* can be best explained by this image :

[](http://2.bp.blogspot.com/-M4Wnr4fZ8CE/U-Nxxj-pTbI/AAAAAAAABwI/pthsD5j4ZtI/s1600/Class+vs+Object+in+Java.gif)

 Here Player is a class which is actually the blueprint of creating players and two players Scooby and Tabby are objects which is created by runtime environment, Java Virtual Machine in this case. Each Player has different value for their attribute, also known as state of Object. For example Scooy's position is 5th and has $341, while Tabby's position is 18th and has $87.  If you still doesn't quite get difference between Class and Object then here is one more example, which you might find more interesting then this one.

[](http://4.bp.blogspot.com/-8-4RPluze6U/U-Nx3jM6TYI/AAAAAAAABwQ/lvJU6CZbKhc/s1600/Class+and+Object+in+Java.gif)

Here CookieCutter is a class which is a blueprint to create objects, the cookies. You can see we have two cookies here, one for you and one for me :)  
  
3) Though [Java is not pure Object oriented language](http://java67.blogspot.sg/2014/03/is-java-pure-object-oriented-programming-language.html), most of things are object in Java, for example primitive variables and operator are not object. In Java, objects are created on a special memory area, known as heap memory. No matter on which scope you create them e.g. locally or globally they are always created in heap space. On the other hand, classes are loaded into another special area of JVM memory, known as permgen space. From Java 8 onward, this space is also known as metaspace. You can create as many object from a class as you want, subject to your heap memory limit, because each object takes some memory. Once all the memory is exhausted, you can not create any more object and JVM will throw [java.lang.OutOfMemoryError: Java Heap Space](http://java67.blogspot.sg/2013/08/guide-of-javalangoutofmemoryerror-java-heap-space-tomcat-eclipse-minecraft-jboss.html) if you further try to create object.  
  
4) Objects are also known as instances in Java programming language, and in JVM class is also represented by an instance of java.lang.Class. On the other hand class is also known as type. A reference variable which holds reference of an object has a type, which denotes what kind of object it can reference. For example in following code ConstructorDemo is name of the class, known as type here, cd is a reference variable of type ConstructorDemo, which means it can either point a ConstructorDemo object or it's child classes. When you create Object using new() operator it automatically class the constructor of that class as well.



5) Class is an abstraction to contain code, mostly attributes and methods which operate on them. On the other hand objects are the real thing which operates on them, but there comes [static methods](http://java67.blogspot.sg/2012/08/can-we-overload-static-method-in-java.html), which belongs to class.  
  
  
That's all on this question about **difference between class and object in Java**. As I said, answers like a class is a blueprint and objects are real things created out of those blueprint is absolutely correct answer, but you must examine further about practical aspect of class and object. If a programmer can differentiate between class and object in code, can give you couple of examples of what class and object are then it's understanding of this important concept is good enough to consider.

#### 3 comments:

* Class and Object are not different but related to each other. You cannot have object without class. Class denies attributes of Object and contains method which can be called upon object. Though Java Class is different than C/C++ structure because you cannot map them directly into memory, JVM does that for you. In C++ you can directly map struct to pointer and read individual values by doing pointer arithmetic; this is not possible in Java. Even JVM can reorder how you define member variable to efficiently utilize a memory page.
* Both class and object is pillar of object oriented programming, which is what Java is. They help you to create flexible software and reduce code duplication. Class defines the blue print from which objects are created. You can create as many objects from a class as you want but each of them will take some memory in heap and you will eventually ran out of memory. Class structure is also loaded into memory, but on different section known as PermGen space prior to Java 8 and now MetaSpace.
* Why use object oriented programming if you can do this things faster in procedural language like C? I love to write procedures, it’s easy, less code to maintain and works like charm. On the other hand you need to write several classes for same thing in Object oriented language like Java. Why would one bother writing so much code if it can be done in 10 lines on procedural way.

Read more: <http://java67.blogspot.com/2014/08/what-is-difference-between-class-and-object-java-programming-oops.html#ixzz3n1mGNe8d>