* When Java is compiled, it is not compiled into platform specific machine, rather into platform independent byte code. This byte code is distributed over the web and interpreted by virtual Machine (JVM) on whichever platform it is being run.
* **Object** is a runtime entity and it’s state is stored in fields and behavior is shown via methods. Methods operate on an object's internal state and serve as the primary mechanism for object-to-object communication.
* A **class** is a blue print from which individual objects are created. A class can contain fields and methods to describe the behavior of an object.
* A class consist of Local variable, instance variables and class variables.
* Variables defined inside methods, constructors or blocks are called local variables. The variable will be declared and initialized within the method and it will be destroyed when the method has completed.
* Instance variables are variables within a class but outside any method. These variables are instantiated when the class is loaded.
* These are variables declared with in a class, outside any method, with the static keyword.
* Constructor gets invoked when a new object is created. Every class has a constructor. If we do not explicitly write a constructor for a class the java compiler builds a default constructor for that class.
* The String class is immutable, so that once it is created a String object cannot be changed. Since String is immutable it can safely be shared between many threads ,which is considered very important for multithreaded programming.