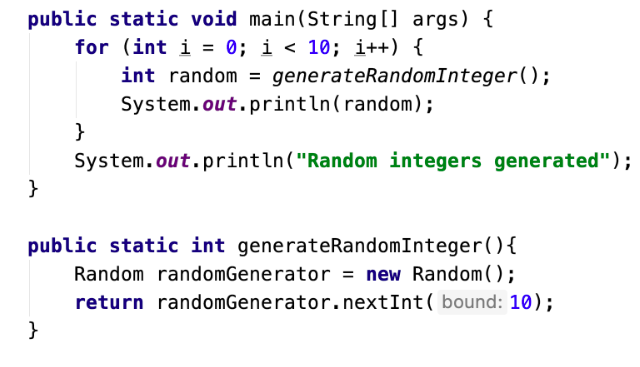
**Repetition**

**Inheritance** is the ability for a superclass to extend its attributes and methods to subclasses. A superclass or parent class is a class that contains one or several attributes and/or methods, that certain subclasses have in common.   
In example: we have a superclass called NPC, this class could contain attributes such as height, age, hair colour and skin colour. These attributes could be inherited by subclasses called female, male, troll, ogre etc. Furthermore, it could have methods such as move, interact, etc.   
In this example it would make sense to have the superclass be an abstract class, since the instantiation of the superclass would never be needed.  
It is worth noting, that a subclass can only inherit from a single superclass and not multiple.

Another form of inheritance is the ability for an interface to be implemented in a class. The empty-bodied methods created in the interface can be implemented as a set of rules for a class to abide by. In the above example, we could have empty methods in our interface called walk, talk, punch. These methods describe something that an NPC per default is able to do, but exact function may vary from class to class.   
Opposite to extending, a class can implement an infinite number of interfaces.



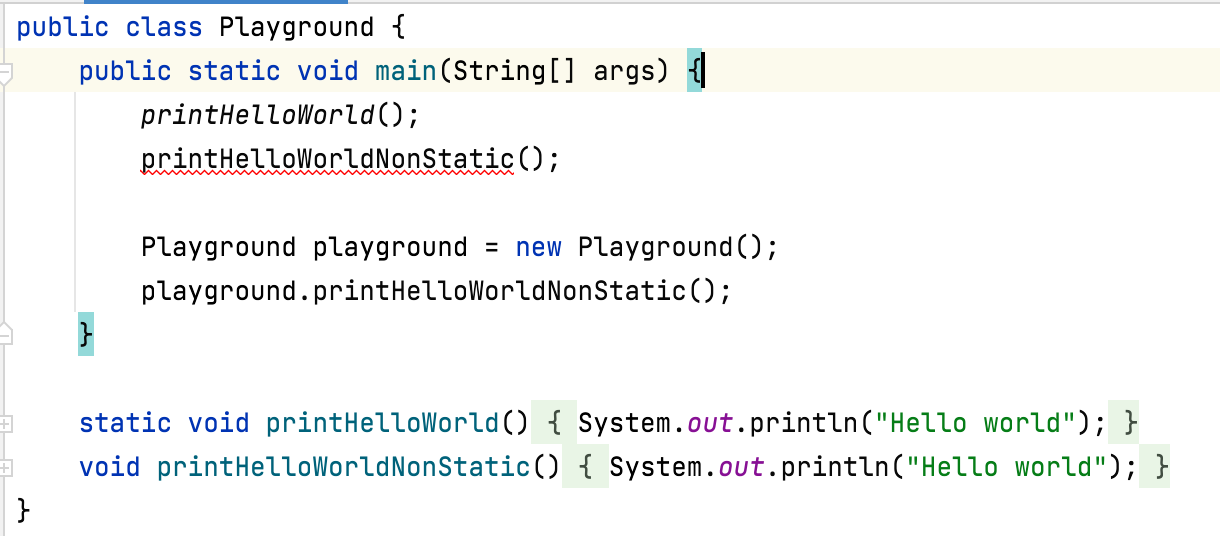
In the code above we have a method called generateRandomInteger. This method creates a new object of type Random, called randomGenerator. The method generateRandomInteger returns an integer, which is determined by the randomGenerators method called nextInt. This method returns a random number from zero and up till the number given as parameter, in this case ten.

The executed code consists of a for loop and a System.out.println command. The for loop iterates ten times due to the fact that it starts at zero and runs while i is less than ten; a value of one is added to i in each iteration effectively stopping the loop after ten iterations.

In each iteration of the loop a variable of type integer is created and assigned a value of 0, 1, 2, 3, 4, 5, 6, 7, 8 or 9 using the above-mentioned method called generateRandomInteger. When the variable has been declared and a value assigned, it is printed using the System.out.println method.

When the loop has looped through all ten iterations, lastly, the string “Random integers generated” will be printed using System.out.println method and the code will finish.

**The static keyword** can be applied to methods, variables and classes. When the static keyword is applied to a method, the method can now be called freely within this class, without the need to instantiate the class firstly.

****

As shown above, the method-call printHelloWorldNonStatic(), which is identical to printHelloWorld() besides the static keyword, computes an error. In order to make use of the method, we have to create an object of type Playground which will be able to call the method.

The static keyword can also be applied to variables, which can be relevant when used for class objects. In an example found online, this can be used to keep track of the number of objects created of said type. In this case, the variable being static is essential, since it belongs to the class itself and not the objects of the classes type.

To summarize, you could say the static keyword makes it so that methods, classes and variables are independent from the classes in which they are instantiated, when applied.