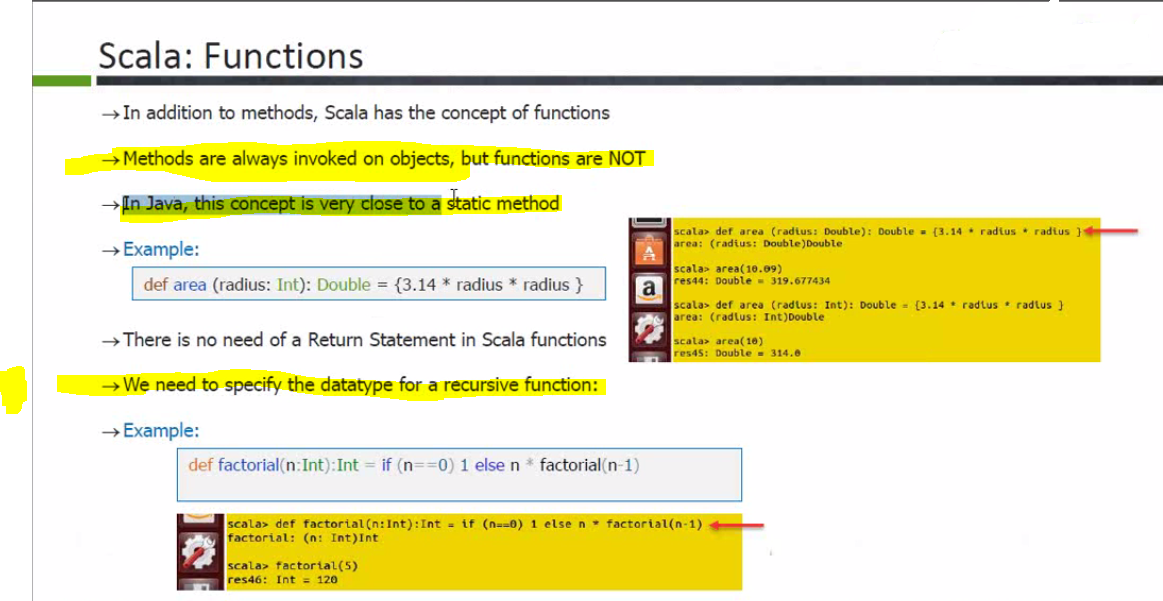
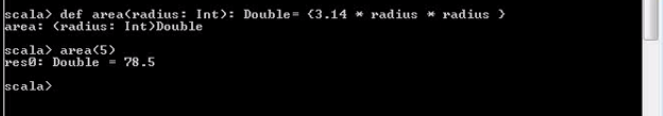
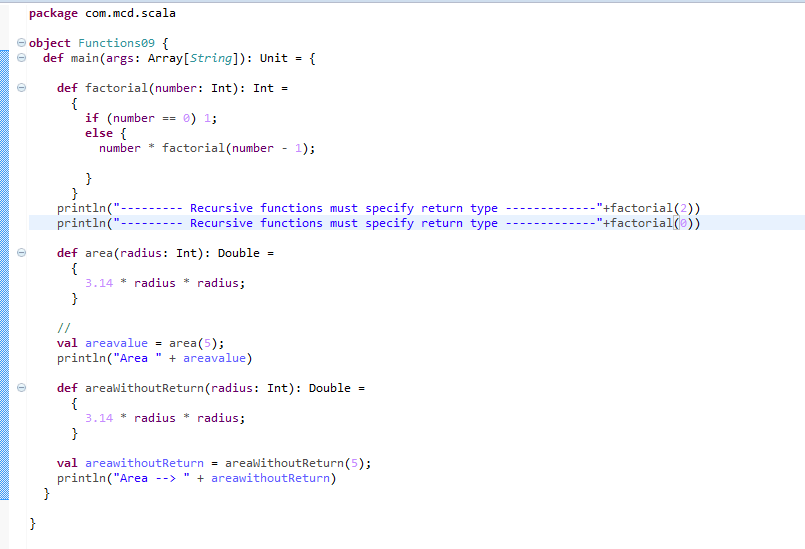
**Scala Functions**





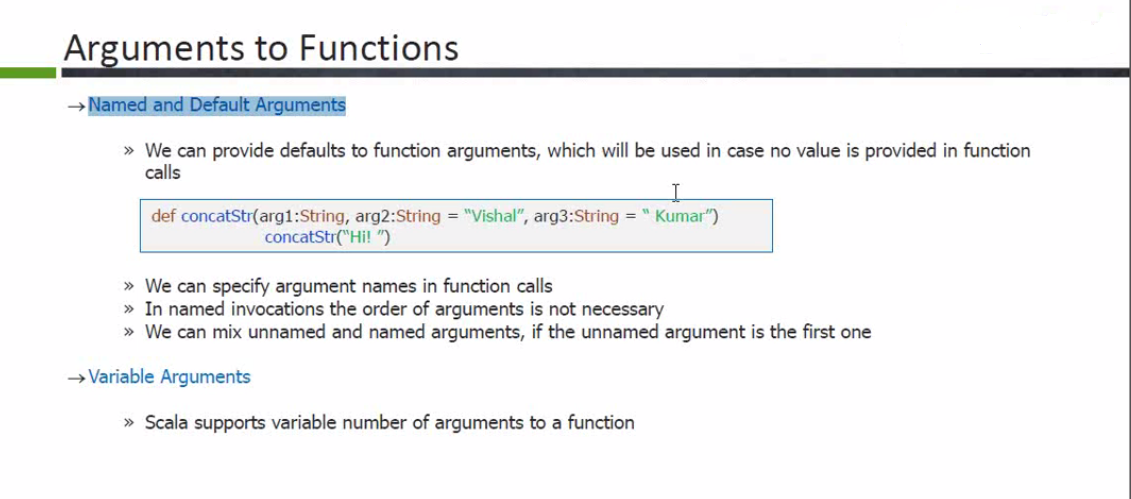


**Methods are invoked on objects whereas functions are independent functionality so they are close to static methods in java (Note they are called on class. So it is only a paradigm which is close to static method in java)**

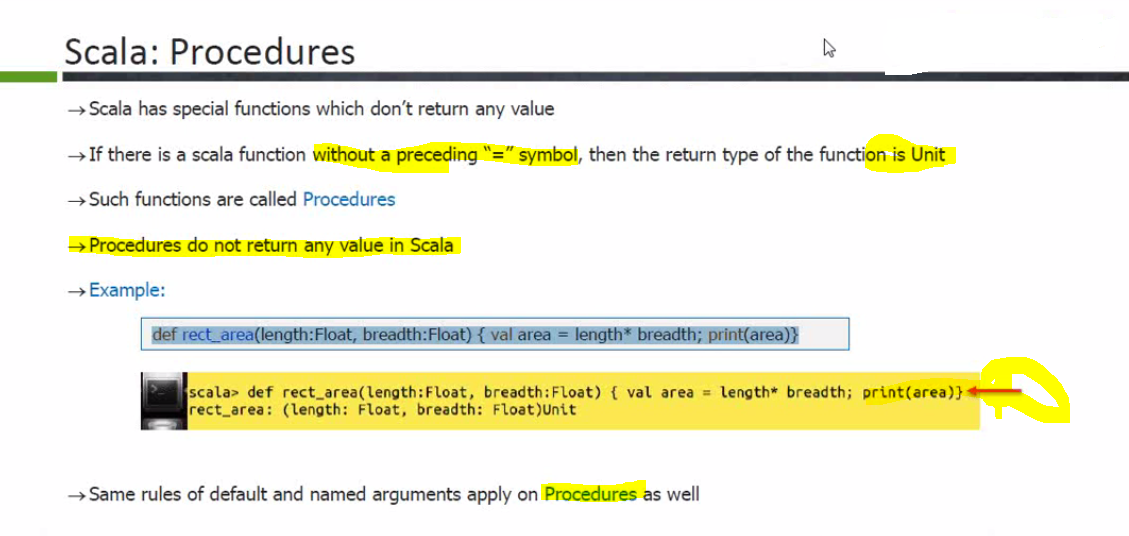
Scala has both functions and methods and we use the terms method and function interchangeably with a minor difference. A Scala method is a part of a class which has a name, a signature, optionally some annotations, and some bytecode where as a function in Scala is a complete object which can be assigned to a variable. In other words, a function, which is defined as a member of some object, is called a method.

A function definition can appear anywhere in a source file and Scala permits nested function definitions, that is, function definitions inside other function definitions. Most important point to note is that Scala function's name can have characters like +, ++, ~, &,-, -- , \, /, : etc.

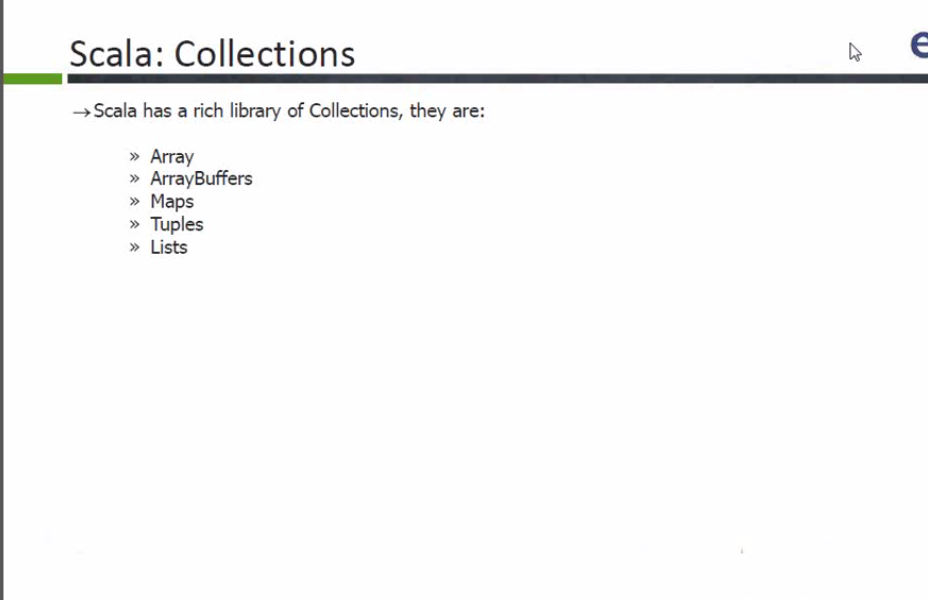
**Scala Functions with Default Values:**

****

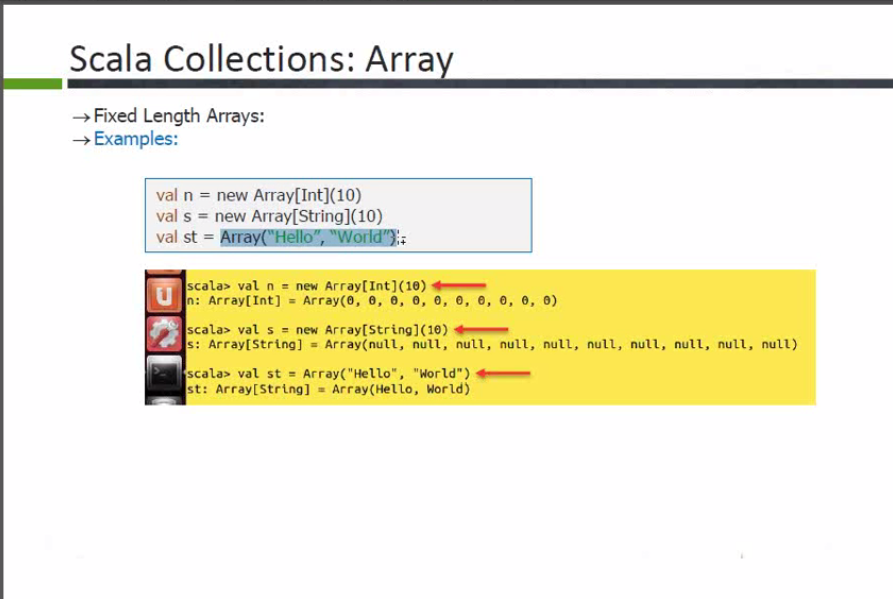
**Scala Procedures.**



**Scala Collections**



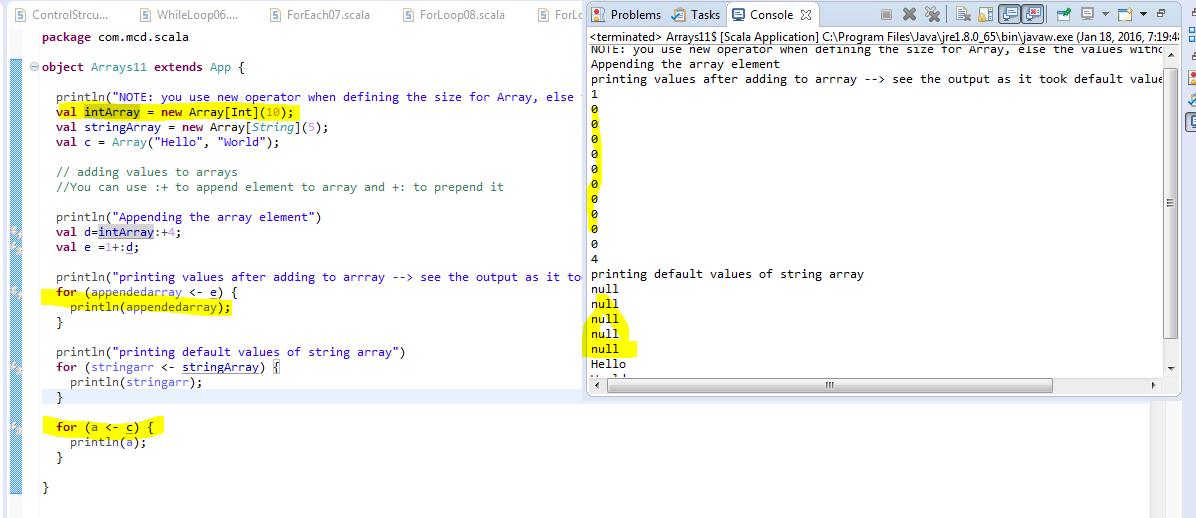
**These collections has 2 packages both mutable and immutable.**

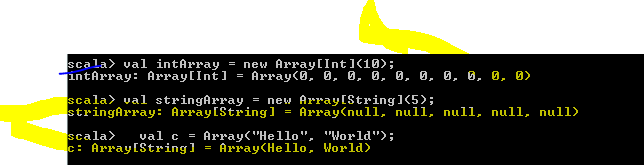


You can use :+ to append element to array and +: to prepend it

**Difference between array and arraylist, array is fixed length where as arraylist is variable length. ArrayBuffer is similar to ArrayList in scala. Array is immutable where as arraybuffer is mutable. So appending and prpending will create new array not update old one.**

**By default, it is immutable package. So to make it mutable you have to import the packages as shown below.**





**You can see the default values for the array. Even if you add, there are default values.**

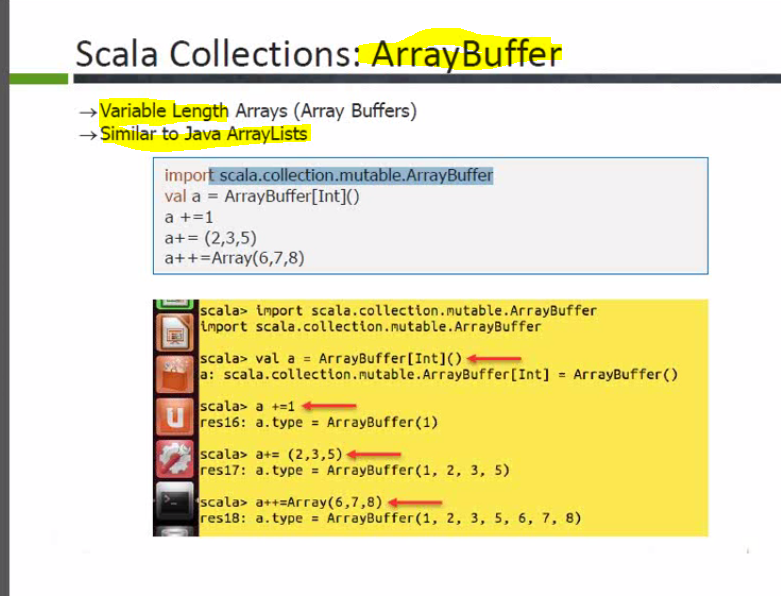
**Appending the array element with :+ or +: will create new array as array is immutable in scala**

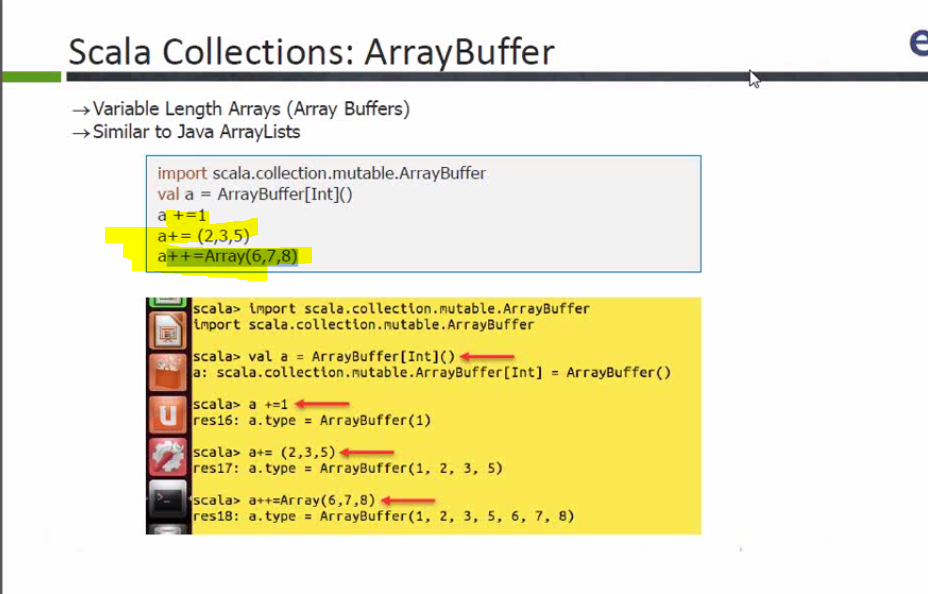
There are a few different ways to create String arrays in Scala. If you know all your array elements initially, you can create a Scala string array like this:

|  |
| --- |
| val fruits = Array("Apple", "Banana", "Orange") |

If you don't know the strings that you want in your array initially, but know the size of your array, you can create it first, then populate it later, like this:

|  |
| --- |
| val fruits = new Array[String](3)    // somewhere later in the code ...    fruits(0) = "Apple"  fruits(1) = "Banana"  fruits(2) = "Orange" |



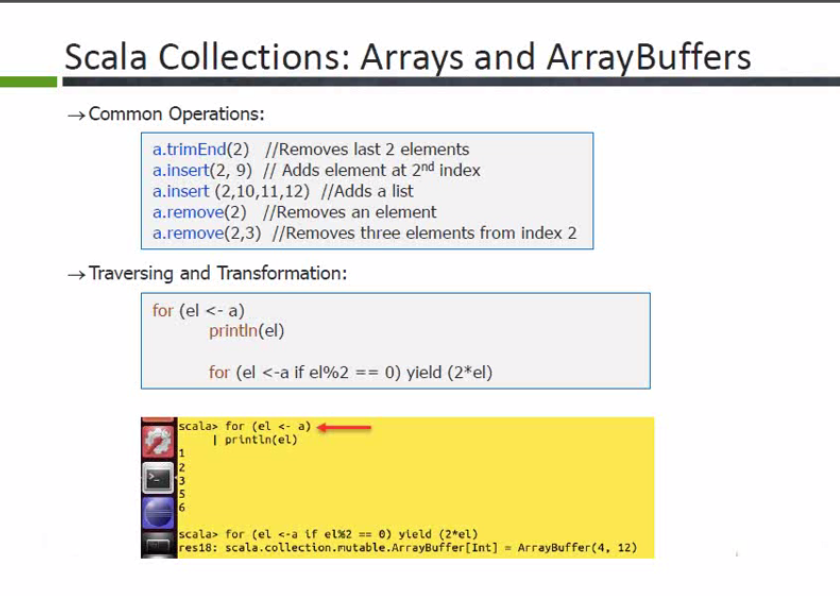


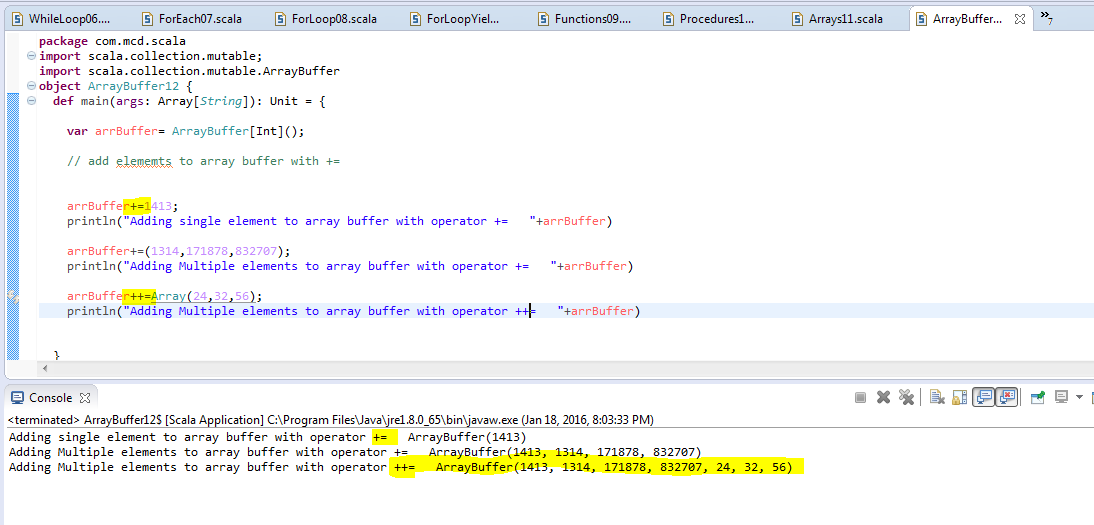
**First 2 statements are just adding elements to list, and 3 rd one is merging array to new array**

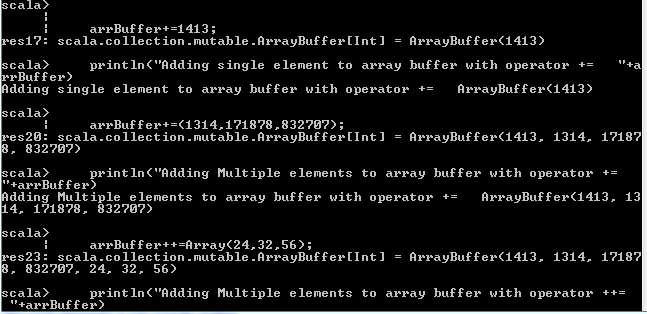
**+= to add element to array**

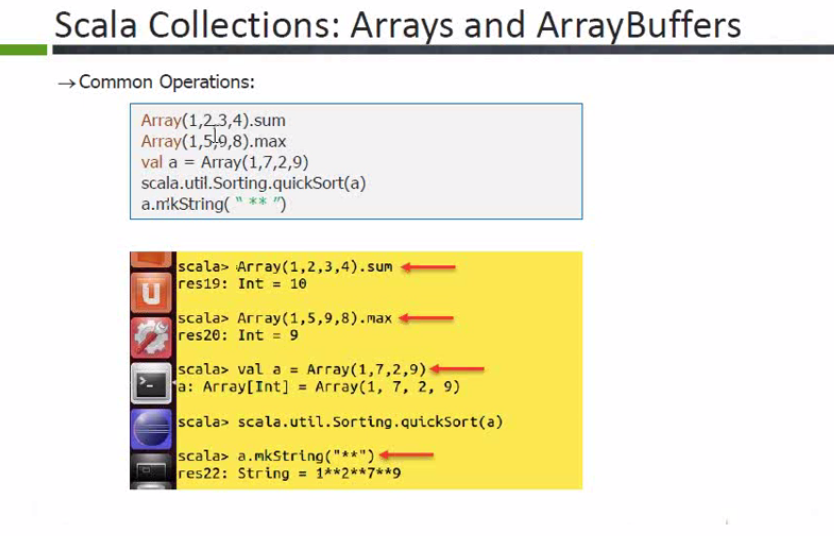
**+= to add multiple elements to array**

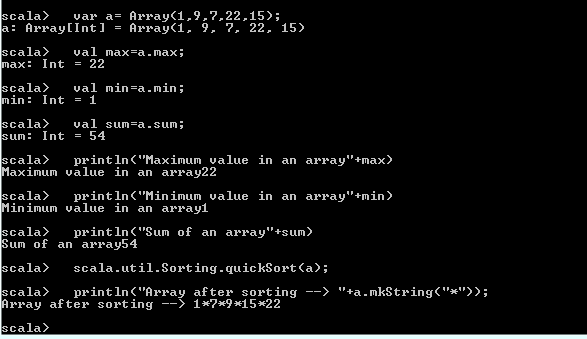
**++= merge array to existing array.**

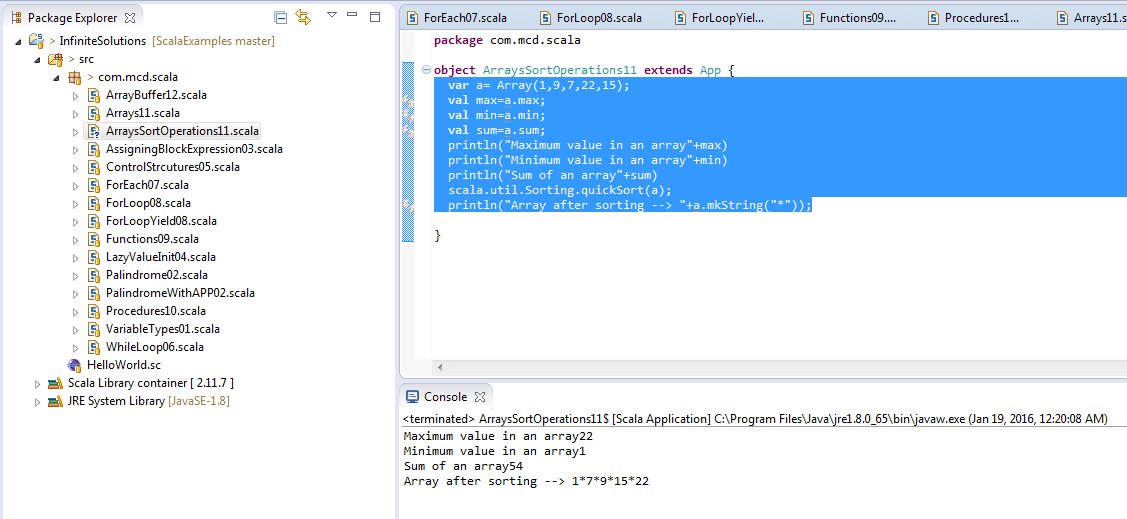




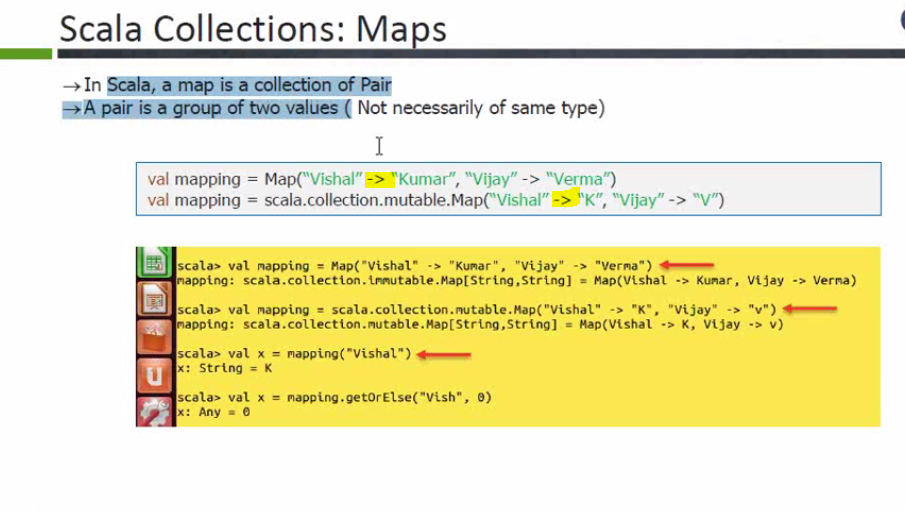


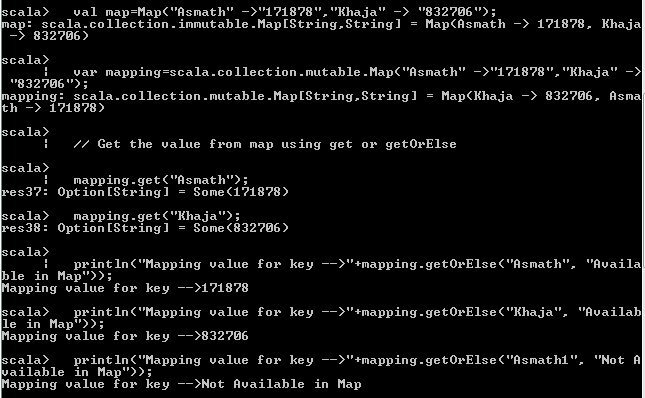


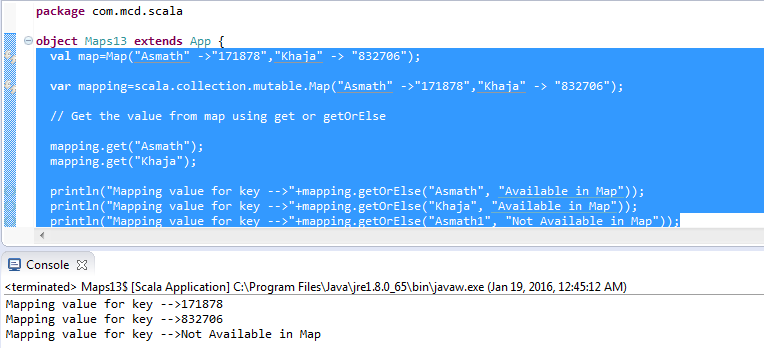




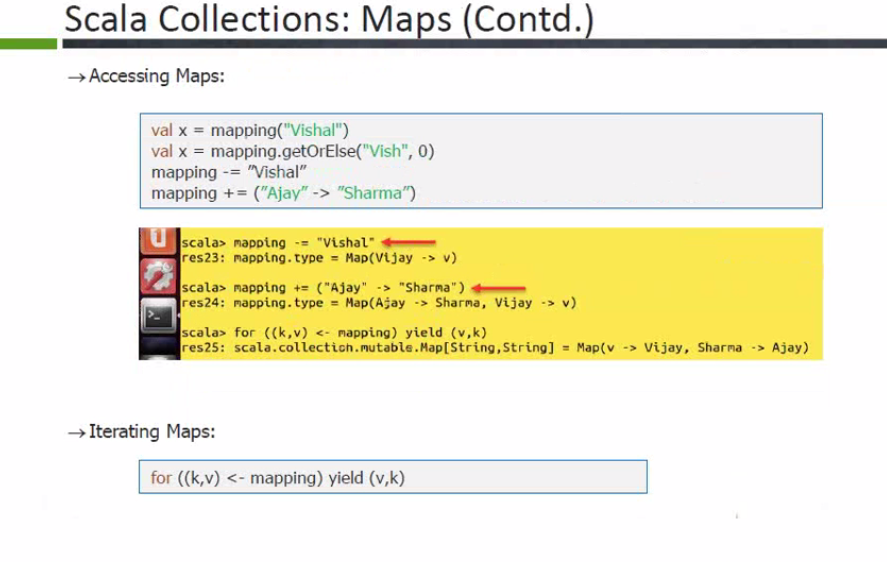
**Maps in Scala -----🡪**

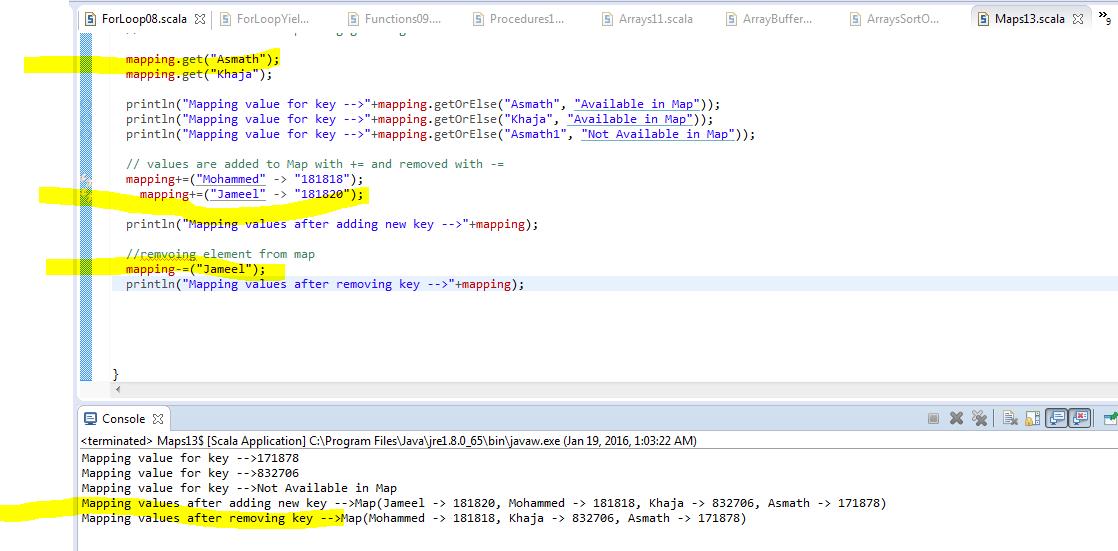


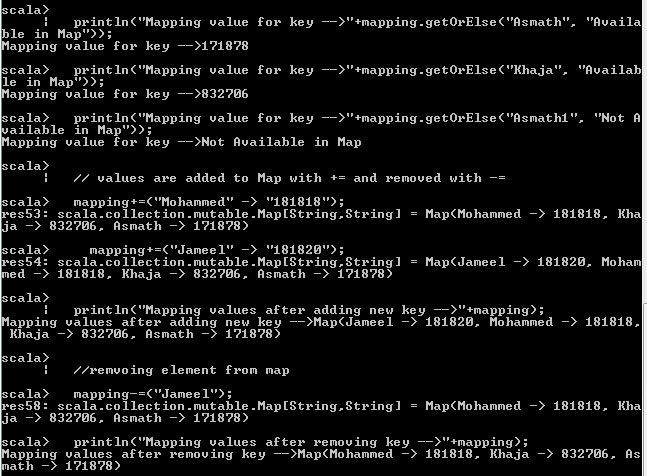


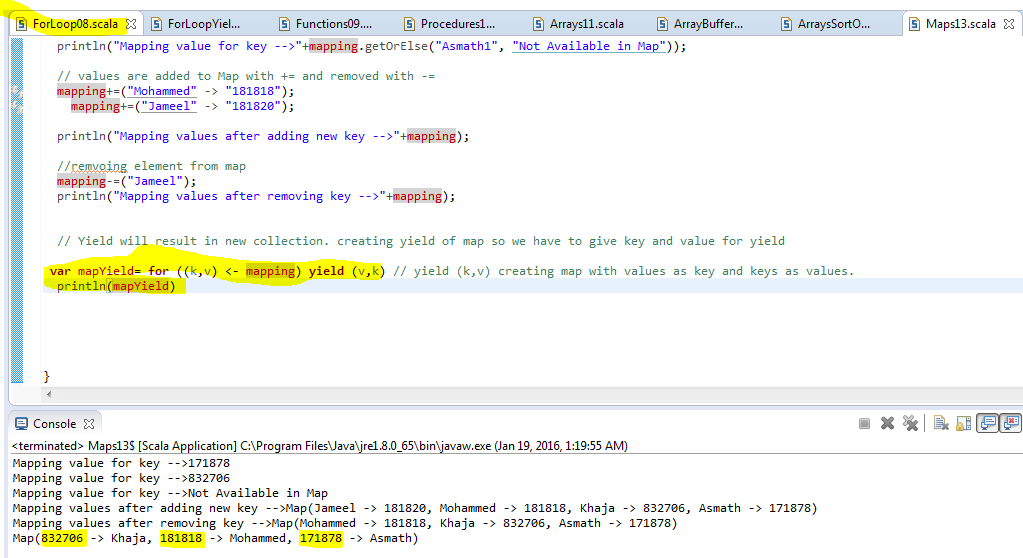


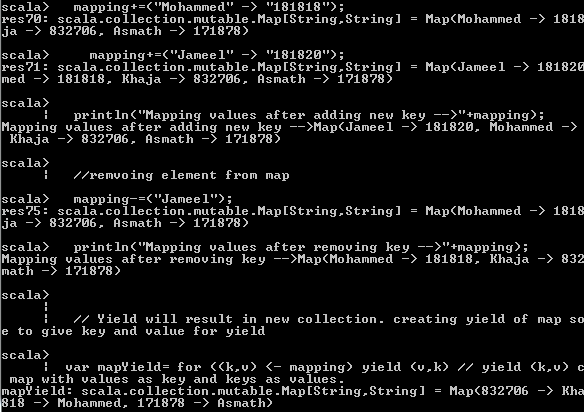
**Add elements in map with += and remove with -=**



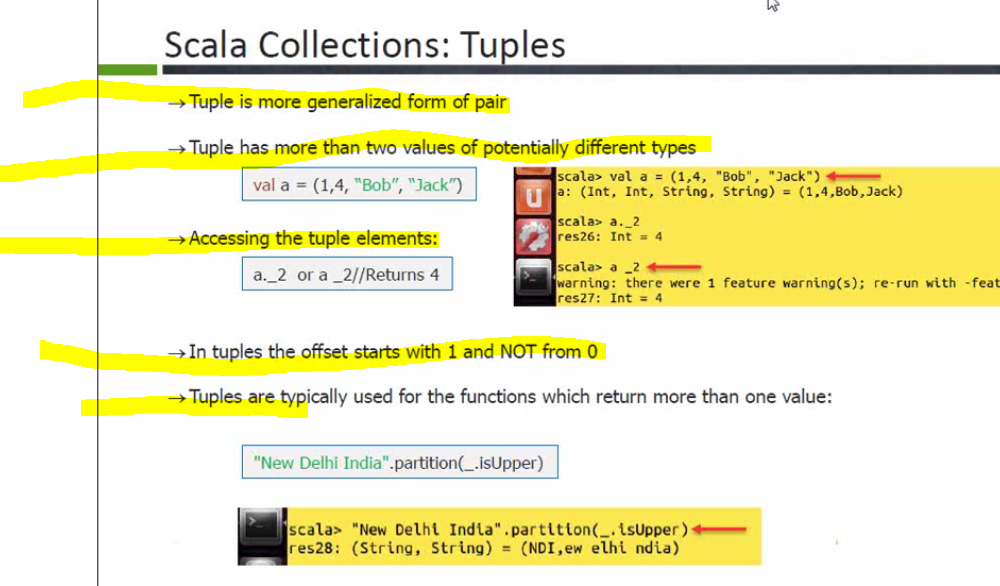








**Tuples in scala 🡪**

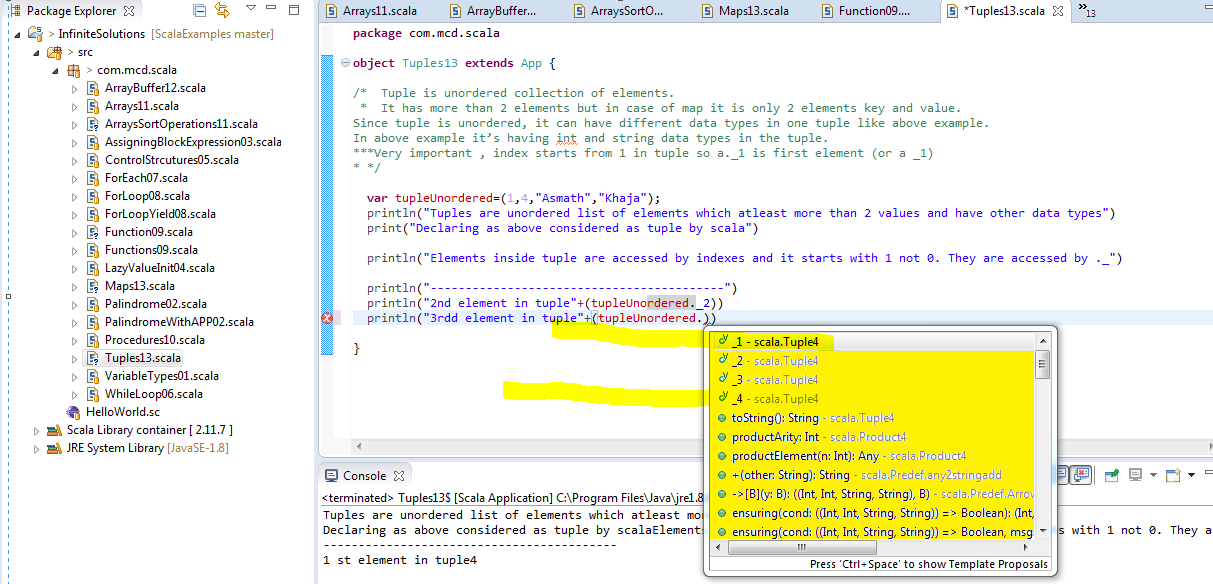


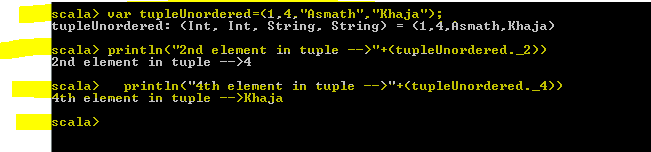
**Tuple is unordered collection of elements. It has more than 2 elements but in case of map it is only 2 elements key and value.**

**Since tuple is unordered, it can have different data types in one tuple like above example. In above example it’s having int and string data types in the tuple.**

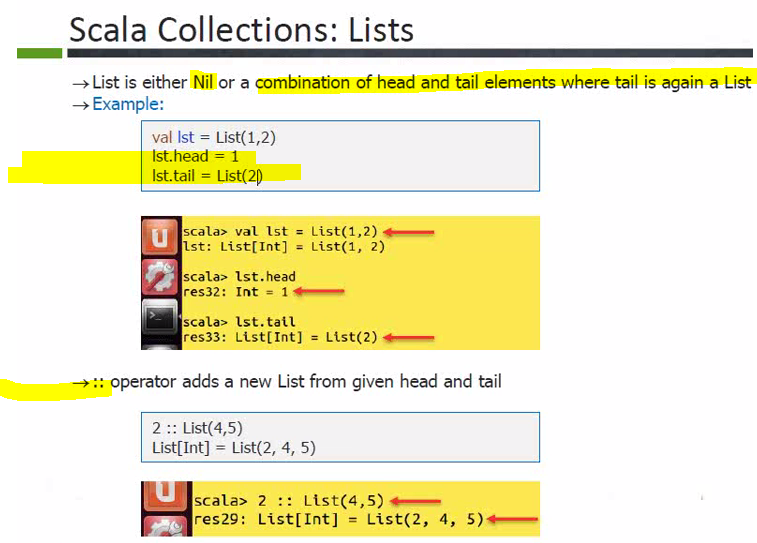
**\*\*\*Very important , index starts from 1 in tuple so a.\_1 is first element (or a \_1)**

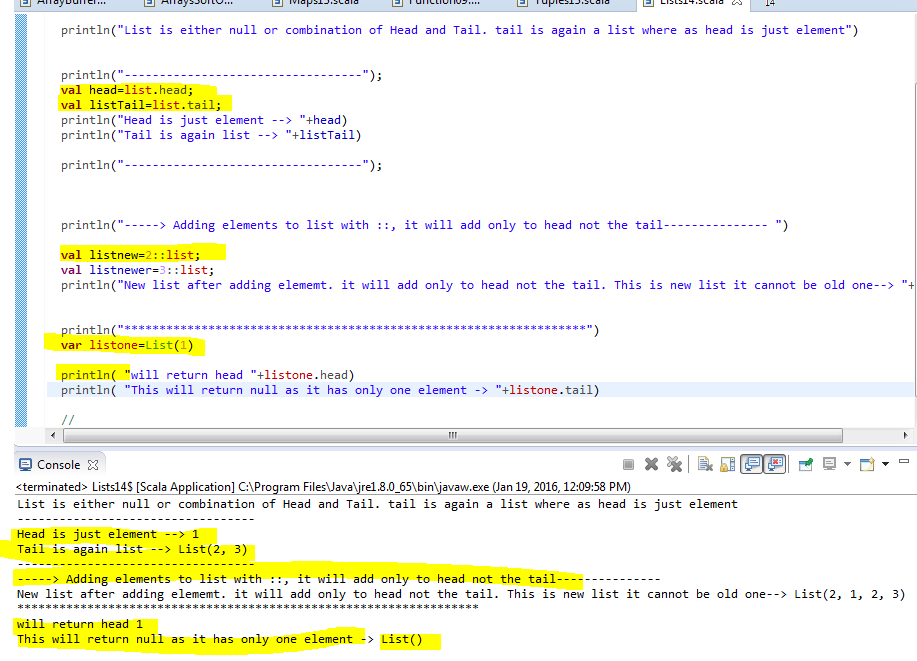
**Hit .space in eclipse and you get the below window with tuple indexes.**

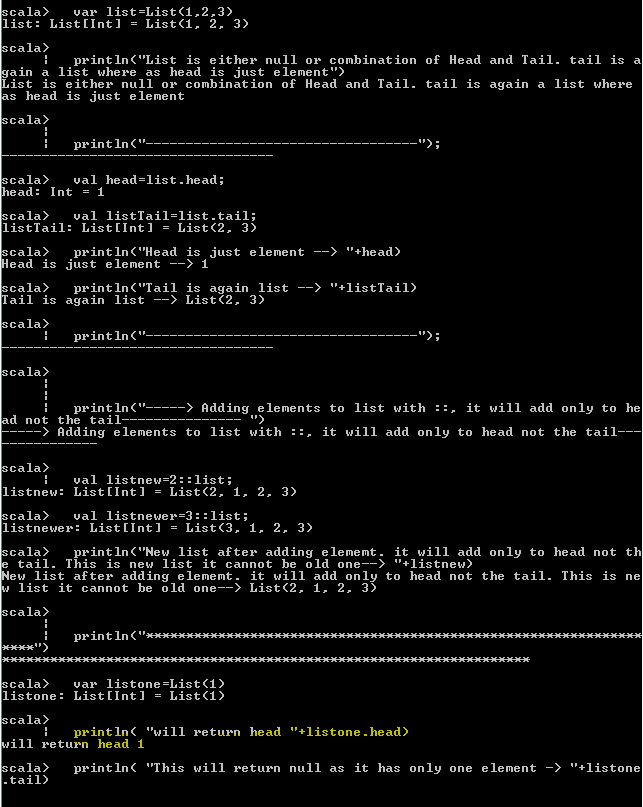


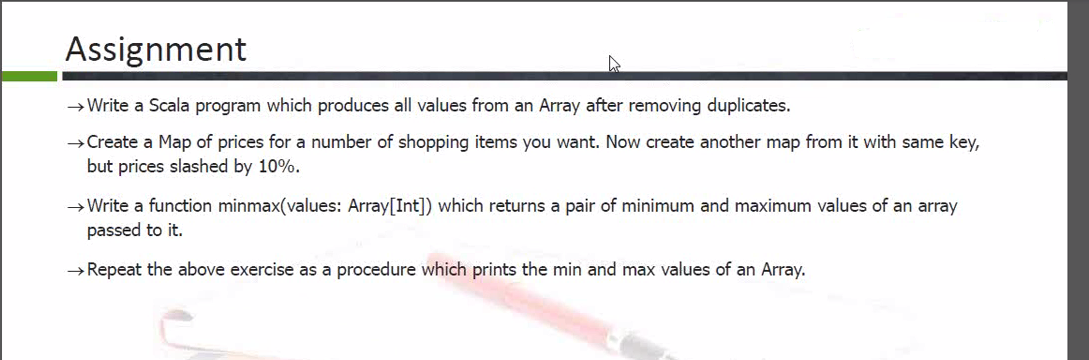


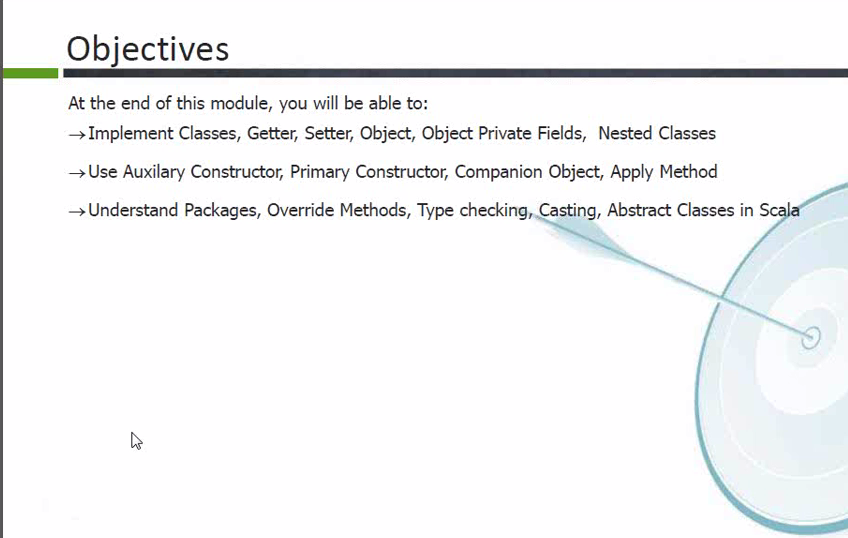
**------------------------------------Lists in Scala --------------------------------------🡪**



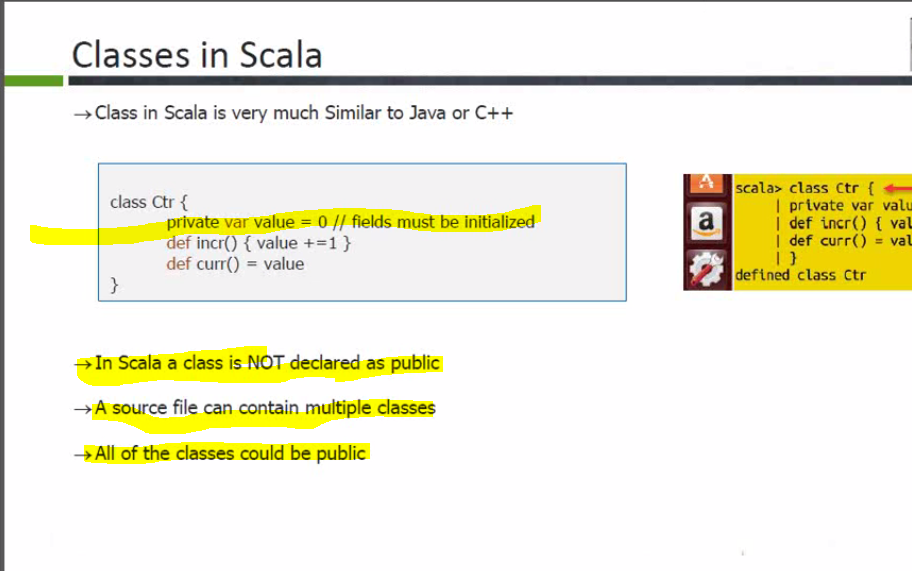




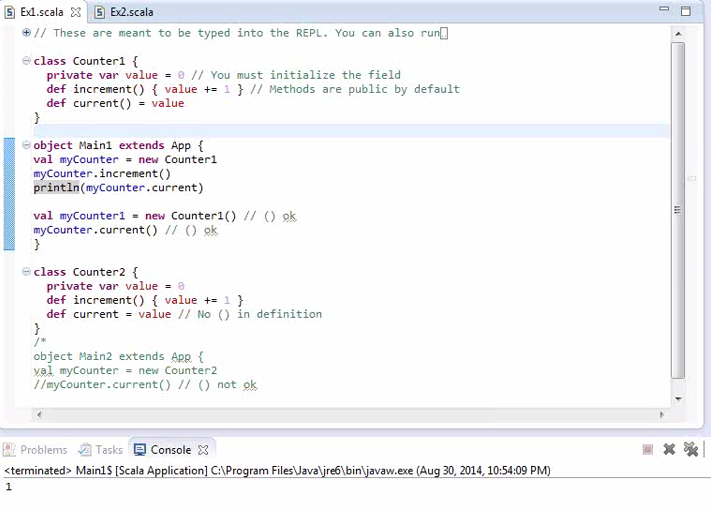
****



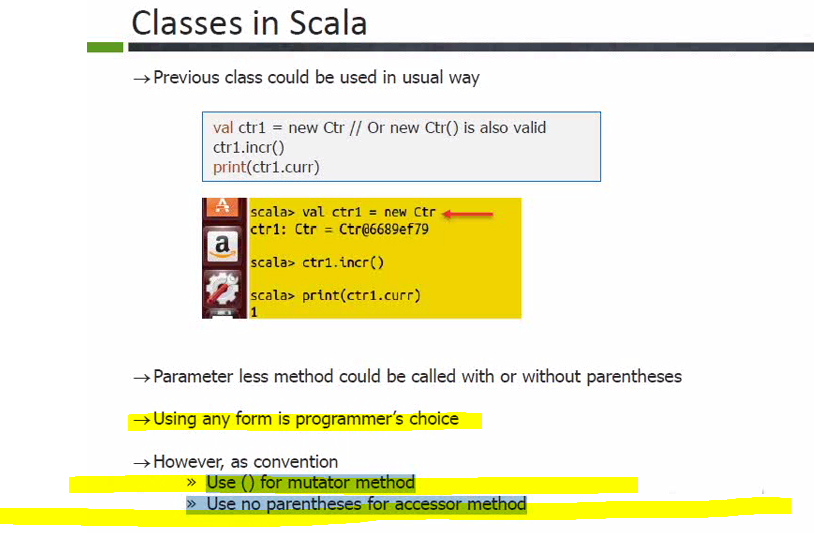
**Classes in Scala**

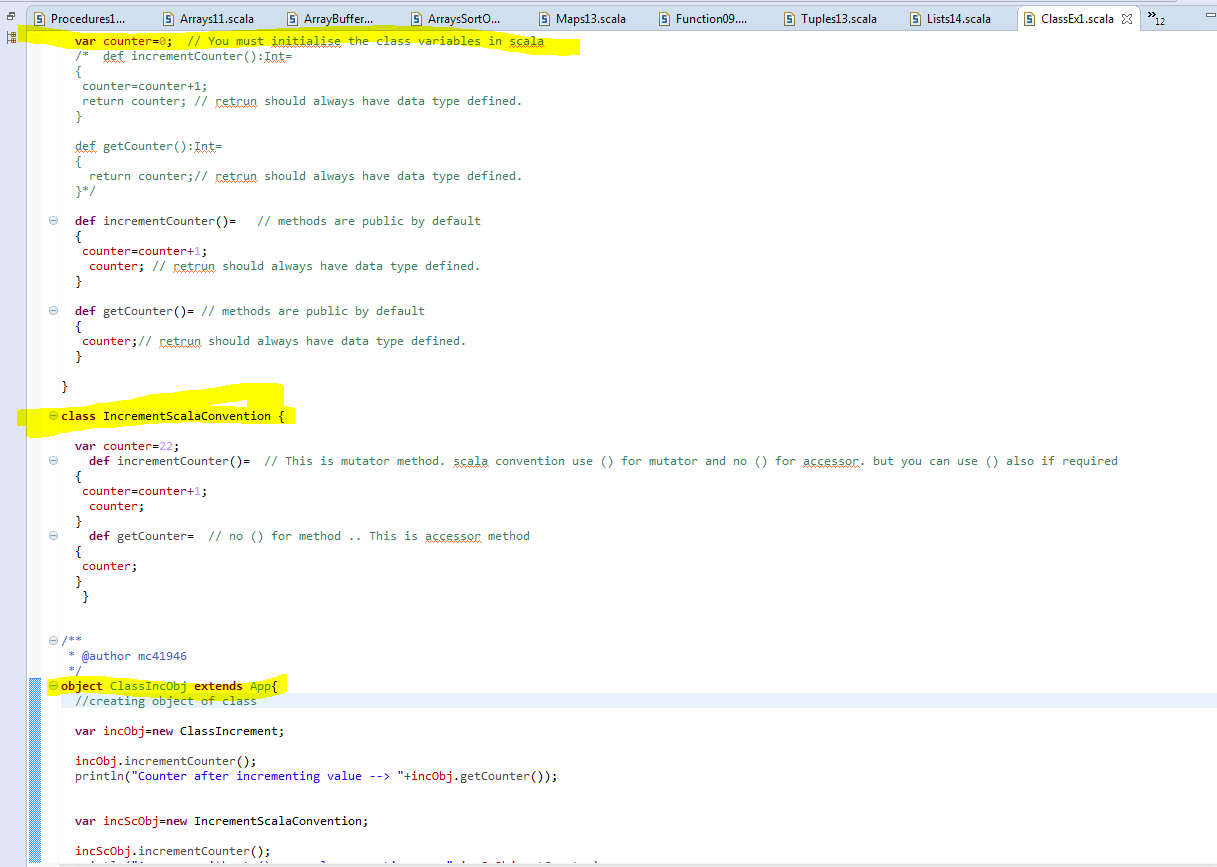


* **Class is not declared as public. Same file can contain multiple classes and they can be public.**
* **All the class level variables should be initialized in scala and not like java**
* **Class file name can be different to the class name. In java class name and file name should be same**

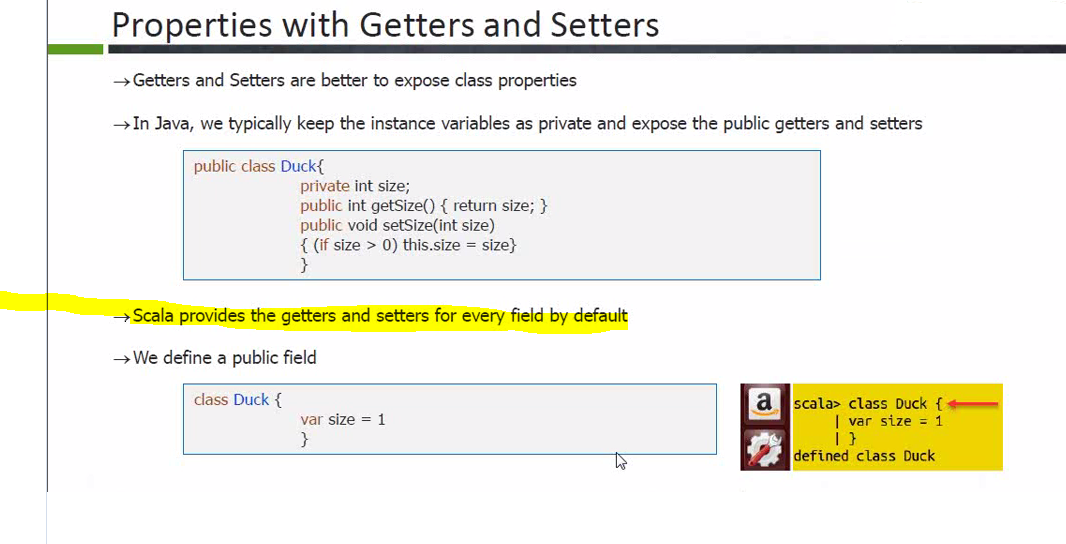


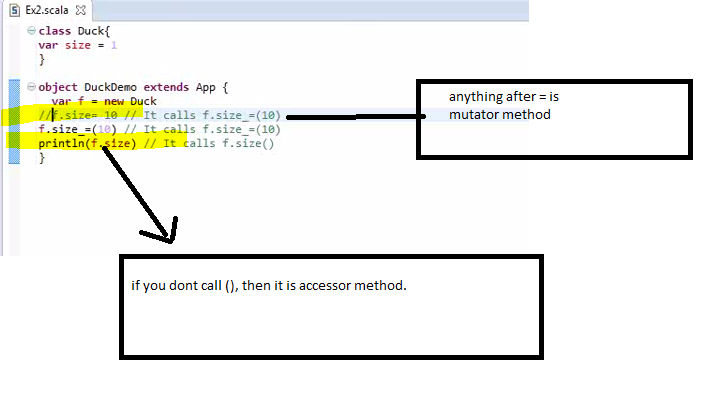
**Note the differences between accessor and mutator. You can write () for both but as scala convention you use without () for accessor and () for mutator.**





**Getter and setters in Scala**



****



**Scala Read only Properties in Classes**





