**JAVA 8**

* **FUNCTIONAL INTERFACES:**
* An interfaces that contains exactly one abstract method is known as Functional Interface.
* Functional Interface can have any number of default, static methods but can contain only one abstract method.
* We can use @FunctionalInterface annotation to mark an interface as functional interface.
* Abstract Method – Has no implementation.
  + Example: public void execute();
* **LAMBDA EXPRESSION:**
  + Lambda expression is an anonymous function.
  + It is a function without name and does not belong to any class.
  + Lambda expression is mainly used to implement Functional Interfaces.
* **LAMBDA VS METHOD:**
* METHOD:
  + Method always belongs to class or object in java.
  + Method has the main parts:
    - Name
    - Parameter list
    - Body
    - Return type
  + Example: public int addition(int a, int b){ return a+b; }
* LAMBDA:
  + Lambda doesn’t belong to class or object.
  + Lambda has the main parts:
    - Parameter list
    - Body
* **FUNCTIONAL INTERFACE:**
* An interfaces that contains exactly one abstract method is known as Functional Interface.
* Functional Interface can have any number of default, static methods but can contain only one abstract method.
* We can use @FunctionalInterface annotation to mark an interface as functional interface.
* Abstract Method – Has no implementation.
  + Example: public void execute();