Functional Interface:

* interface which has only one abstract method.
* Also known as SAM (Single Abstract Method).
* @FunctionalInterface can be used to indicate the compiler that this is a functional interface and be throw error if anyone breaks the rule, and forces us to use only abstract method.
* We can have default methods, and static methods.
* Special to mention we can have methods of object class.
* One never heard special case, if one funcInt implements another funcInt, and both has method with same signature (overridden) then it is perfectly fine and does not violate funcInt rules.

Lambda expression:

* Lambda expression is a way to implement the function interface.
* There are 4 types of built in functional interfaces under java.util.function
  + Consumer -> accept(T t) – one input - return no result
  + Supplier -> T get() – no inputs - returns one result
  + Predicate -> boolean test(T t) – one input, one Boolean result
  + Function -> R apply(T t) – one input, one output