1) In java, all the primitives are passed by value by default. When you have a collection or something similar, you pass the referrence of this collection as a value. Also for the primitives, elements are being passed as value type referrences. They are refferrences but we pass the value of refferrences.

   So in practice, if we assign the object to a new value, it will change outside of the function as well. However, if we assign the referrence to a new object, referrence will just start to point the new object, and we will totally lost the control of the passed value, and it will not change outside of the function.

2) Immutability means an object is unable to change. If we want to change an object, we basicly destroy the previous (if there is not another referrence or pointer pointing to this object) one and create a new one in the changed shape. In this process, other referrences that points to this object will stay with older object value. If there is no other referrence pointing to the old object, it will be deleted.

   to create a immutable class in java, all the fields and class itself must be initialized as final. There should be no setters. If any new assigment operator happens, current values copied to a new created object. All the getters method and other functions must be created by getters methods.

3) Library is consist of well defined functions, and with using these functions we can perform various operations. Framework is a complex system that can contains multiple libraries or another necessary operators. In these systems, we can either define functions, or can give the framework the power of calling other functions or creating objects.

4) Garbage collection is simply the process of destroying unnecessary elements in a software. In a software, we time to time create and use objects, and after a while we might no longer needed these objects. In java, these objects are deleted automaticly by garbage collecter (GC). GC in java works like this; time to time it runs on JVM and checks through the past and detects the elements that are no longer needed or have no access bu user and free these memories. This operastion runs on JVM and it requires usage of RAM. This is the main reason why java applications consume lots of memory and energy.

5) in java memory leak is only possible when an unnecessary big amount of data has been created but never being used. Since GC will not keep track of it’s usage, it will keep itbecause of it’s existed access. So, we should be careful about big amount of data especially. Also, unclosed resources can be painful since stream will be open and memory allocated for these events will not be freed.

6) Oracle intends to release LTS version every 2 years. On the other hand in practice, almost every year a java version has been released so far.

7) Stack is a memory that is being controlled and optimized by computer. So far it can control the initial objects and variables that can be used in functions. This memory is fast and optimized but limited. On the other hand, heap is being used for dynamic memory allocations. It is not ordered, less efficient but not limited. In java it has been used for global scope variables and dynamic memory allocations. Stack only contains primitive types or the referrences that points to these types. Heaps can contain any type.

8) OpenJDK is an open source implementation of java and it is free for everyone. It is less efficient then OracleJDK. OracleJDK is a paid implementation of Oracle for enterprises. It has better performance then OpenJDK.

9) I quite did not understand the @FcuntionalInterface and functioanl interfaces generally. So I will skip the questions number 9 and 10. I will update after the class when I learn it properly.