Umut Yıldız

1) Object Oriented Programming provides modular and reusable coding. We can solve and map domain problems using classes easily. We can remove redundant code using inheritance. We can solve the data hiding problem which is important for some architectural designs with oop. We can convert the real problem objects to code easily. Java, C++, C#, Python are some major OOP languages.
2) For abstract class, child classes of root abstract can directly use methods of abstract class or they can override and implement these methods but this situation is invalid for interface. Programmers can use interface as a reference for class which implements this interface but can not create object of abstract class and interface. Abstract class has a is-a relationship but interface has has-a or can-do relationship. A class can extend only one abstract class but a class can extends multiple interfaces. Interface doesn't have constructor on the other hand abstract class may have constructor.
3) Hashcode method always return same integer for same objects and equals method is comparator method for objects. If we do not override, we use the default implementation in the object class and in this method only the same memory address value is checked, but when we override, we can compare not only the memory address but also the properties.
4) Java doesn't support multiple inheritance. For example, we want to inherit A class from B class and C class. We can solve this problem using interface in java.
5) There may be unused or unreachable objects in memory, and the garbage collector will clean them and we don't have to worry about memory while creating new objects. It runs automatically and we cannot predict when garbage collector works.
6) When we use static keyword for a class or etc., it is created only once in heap. We can use static for class, method, variable or block. It is accessible without reference of any object or instance of any class.
7) Immutability means as cannot changeable that is an object is created only once and we cannot change its content. We use immutability because it makes easier to parallelize our program without conflicts amount objects. We can use immutability for class and field. We can use for class which is marked final keyword, have no setter methods and have all fields which are marked as final and private.
8) Composition and Aggregation are types of relationship between objects. Child classes are unique for parent class in Composition. They cannot be created with themselves but parent class can be generated without sub classes in Aggregation. For example, there are Team, Player and Organ classes. There is a aggregation relationship between Player and Team because Player can be generated with itself but there is a composition relationship between Organ and Player because Player cannot be generated without Organ.
9) For example, the methods etc. contained in a class should be tightly connected to each other. This is an example of high cohesion. But dependencies and relationships between classes should be loose. This is loose coupling.
10) Stack is a memory place which holding things with immutable memory size. For example, primitive types are held in stack but things like objects whose memory space is not strictly defined are kept in the heap. Access time of Stack is faster than Heap. Stack is cleaned automatically but heap is not.
11) It is an event that disrupts the normal flow of the program during operation as an error. There are three types of exception as Checked (ex. FileNotFoundException), Error, Runtime Exception (related logic mistakes). Error and Runtime exceptions are unchecked exceptions.

12) The main idea of clean code is easy to change code and easy to understand and read code. When someone else takes a look at the code we have written, it is easy to understand the logic and working flow of the codes, why the defined variables are used, why an expression is used, or simply what a method does in general.
13) After defining a static method with a signature in the parent class, when we define a method in the sub class with the same signature, we actually hide it because we will overwhelm the method in the parent class.
14) Polymorphism is that ability for same object to take different forms but abstraction is related security and reusability.