

HW#1 – Baran AYDIN

1 – Why we need to use OOP ? Some major OOP languages ?

We use OOP because of its 4 most benefits;

- a) Modularity for easier troubleshooting
- b) Reuse of code through inheritance
- c) Flexibility through polymorphism
- d) Effective problem solving

Major OOP languages: C++, C#, Java, Python, Ruby, Swift

2 – Interface vs Abstract class ?

The interface is a blueprint that can be used to implement a class. The interface does not contain any concrete methods (methods that have code). All the methods of an interface are abstract methods.

An interface cannot be instantiated. However, classes that implement interfaces can be instantiated. Interfaces never contain instance variables but, they can contain public static final variables

A class which has the abstract keyword in its declaration is called abstract class. Abstract classes should have at least one abstract method without a body. It can have multiple concrete methods.

Abstract classes allow you to create blueprints for concrete classes. But the inheriting class should implement the abstract method.

Abstract classes cannot be instantiated.

3 – Why we need equals and hashCode ? When to override ?

“==” operator compares the two objects on their physical address. That means if two references are pointing to same object in the memory, then comparing those two references using “==” operator will return true. For example, if s1 and s2 are two references pointing to same object in the memory, then invoking **s1 == s2** will return true. This type of comparison is called **“Shallow Comparison”**.

equals() method, if not overridden, will perform same comparison as “==” operator does i.e comparing the objects on their physical address. So, it is always recommended that you should override equals() method in your class so that it provides field by field comparison of two objects. This type of comparison is called **“Deep Comparison”**.

In java.lang.String class, equals() method is overridden to provide the comparison of two string objects based on their contents. That means, any two string objects having same content will be equal according to equals() method. For example, if s1 and s2 are two string objects having the same content, then invoking **s1.equals(s2)** will return true.

hashCode() method returns hash code value of an object in the Integer form. It is recommended that whenever you override equals() method, you should also override hashCode() method so that two equal objects according to equals() method must return same hash code values. This is the general contract between equals() and hashCode() methods that must be maintained all the time.

In java.lang.String class, hashCode() method is also overridden so that two equal string objects according to equals() method will return same hash code values. That means, if s1 and s2 are two equal string objects according to equals() method, then invoking **s1.hashCode() == s2.hashCode()** will return true.

4 – Diamond problem in Java ? How to fix it?

Diamond problem (deadly diamond of death) is that a class try to extend two different classes with same properties. Compiler can not resolve property of which super classes should be used.

Java does not support multiple inheritance because of Diamond problem.

The solution to the diamond problem is interfaces since interfaces mandates to override its properties.

5 – Why we need Garbage Collector ? How does it run ?

Garbage collector is the process of managing memory, automatically. It finds the unused objects that are no longer used by the program and delete or remove them to free up the memory.

Java provides two methods System.gc() and Runtime.gc() that sends request to the JVM for garbage collection. These are controlled by JVM and can not be forced to run.

6 – Java 'static' keyword usage ?

The static keyword in Java is used for memory management mainly. We can apply static keyword with variables, methods, blocks and nested classes.

The static variable can be used to refer to the common property of all objects (which is common for all object), for instance: the company name of employees, college name of students, etc.

The static variable gets memory only once in the class area at the time of class loading.

7 – Immutability means ? Where, How and Why to use it ?

Immutable objects don't change their internal state in time, they are thread-safe and side-effects free. Because of those properties, immutable objects are also especially useful when dealing with multi-thread environments. In Java final keyword is used for immutability.

8 – Composition and Aggregation means and differences ?

Composition is a “belongs-to” or “has-a” type of relationship. It means that one of the objects is a logically larger structure, which contains the other object. In other words, it's part or member of the other object.

Aggregation is also a “has-a” relationship. What distinguishes it from composition, that it doesn't involve owning. As a result, the lifecycles of the objects aren't tied: every one of them can exist independently of each other.

Association is the weakest relationship between the three. It isn't a “has-a” relationship, none of the objects are parts or members of another. Association only means that the objects “know” each other.

9 – Cohesion and Coupling means and differences ?

Cohesion is the degree to which the elements inside a module belong together. A module could be a class or a package or even a microservice. Simply put, it means “the code that changes together, stays together”.

Coupling is the degree of interdependence between software modules. A module could be a class or a package or even a microservice. Effectively, the coupling is about how changing one thing required change in another.

10 - Heap and Stack means and differences ?

A stack is a special area of computer's memory which stores temporary variables created by a function. In stack, variables are declared, stored and initialized during runtime.

The heap is a memory used by programming languages to store global variables. By default, all global variable are stored in heap memory space. It supports Dynamic memory allocation.

The heap is not managed automatically for you and is not as tightly managed by the CPU. It is more like a free-floating region of memory.

11 – Exception means ? Type of Exceptions ?

In Java “an event that occurs during the execution of a program that disrupts the normal flow of instructions” is called an exception. This is generally an unexpected or unwanted event which can occur either at compile-time or run-time in application code.

There are two types of exceptions in Java; built-in exceptions and user defined exceptions.

12 – How to summarize ‘clean code’ as short as possible ?

Clean code stands for coding in set of particular rules such as naming, comments, formatting, in order raise human readability and also have a standard of code.

13 - What is the method of hiding in Java ?

Encapsulation is a protective shield that prevents the data from being accessed by the code outside this shield. In addition to encapsulation there are two more way of hiding in Java;

Variable Hiding:

Variable hiding happens when we declare a property in a local scope that has the same name as the one we already have in the outer scope.

local variables – declared in a piece of code such as methods, constructors, in any block of code with curly braces

instance variables – defined inside of a class and belong to the instance of the object

class or static variables – are declared in the class with the static keyword. They have a class level scope.

Method Hiding:

Method hiding may happen in any hierarchy structure in java. When a child class defines a static method with the same signature as a static method in the parent class, then the child's method hides the one in the parent class.

14 - What is the difference between abstraction and polymorphism in Java ?

Abstraction allows a programmer to design software better by thinking in general terms rather than specific terms while Polymorphism allows a programmer to defer choosing the code you want to execute at runtime.

Another difference between Polymorphism and Abstraction is that Abstraction is implemented using abstract class and interface in Java while Polymorphism is supported by overloading and overriding in Java.

Though overloading is also known as compile-time Polymorphism, method overriding is the real one because it allows a code to behave differently at different runtime conditions, which is known as exhibiting polymorphic behavior.
