First Week Homework

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

➤ Project will be more organized, you know exactly where to look when something goes wrong. This causes higher productivity and low maintenance cost.

➤ With abstraction, project will be more secure and datas will be only editable in the related class.

➤ You don't have to rewrite codes. You can reuse your code as many times as needed with Inheritance.

➤ You can extend your code without rewriting it. Through polymorphism you can add on the existing code with "method overriding".

➤ It's easier to understand, reuse and maintain.

➤ Most popular OOP languages are Java, C++, C#, Python and JavaScript.

2 – Interface vs Abstract class?

Interface

➤ There is no code inside the methods and only method definitions.

➤ Access modifiers are not used. Everything is considered public.

➤ Multiple inheritance can be used

Abstract class

➤ Used for gather the objects that have in common.

➤ Used in Is-A relationships and extended class takes all the features.

➤ Methods and variables can be defined.

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

➤ With the Equals method, we get a boolean return that two objects are equal to each other.

➤ If two objects are equal according to equals() method, then their hash code must be same.

➤ But if two objects unequal according to equals() method, their hash code are not required to be different.

➤ So we should use the equals method first and then the hashCode() method. But there is something we should know.

➤ If we are not planning to use the class as Hash table key, then our program will not throw any exceptions.

➤ If we are going to use a class as Hash table key, then it's must to override both equals() and hashCode() methods.

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

➤ Java does not allow multiple inheritance where one class can inherit properties from more than one class.

➤ This is known as Diamond problem.

➤ Because of the interfaces can do multiple inheritance, we can achieve Diamond problem by using interfaces and default methods.

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

➤ Understanding Garbage Collector means writing better and more effective Java applications.

➤ Automatically handles the deletion of unused objects or objects that are out of reach to free up vital memory resources.

➤ The space occupied by unused/non-referenced objects are detach from memory and cleared from memory.

➤ The mechanism that performs this operation is called the Garbage Collector.

6 – Java ‘static’ keyword usage?

➤ The static keyword in Java is used for memory management mainly.

➤ We can use static keyword with variables, methods, blocks and nested classes.

➤ It makes your program memory efficient.

➤ Static variable will get the memory only once, it will retain its value.

➤ In static method, we don't have to create instance of a class. It's belongs to the class rather than the object of a class.

➤ The static method can't use non-static contexts.

➤ this and super keywords can't be used in static methods.

➤ Static block is executed before main method, used to initialize the static data member.

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

➤ Immutability means that we can not change its content.

➤ In Java, immutable class must be initialize as final so child classes can't be created.

➤ Datas in this class must be declared as private final because direct access is not allowed and we can't change the value after the creation.

➤ Must have getter methods and they can't have setter methods or any method that can change the value of the datas.

➤ In immutable class, data won't change. Once verified, it will be valid. Better cacheability.

8 – Composition and Aggregation means and differences?

➤ They are two types of association which is used to represent relationships between two(parent-child) classes.

➤ In composition, there is strong relationship between these two classes. Parent class owns the child class. In UML diagram it is denoted by filled diamond.

➤ In aggregation, there is weak relationship between these two classes.

➤ There is Has-A relationship, which doesn't imply ownership. In UML diagram it is denoted by empty diamond.

9 – Cohesion and Coupling means and differences?

➤ Cohesion refers to what the class (or module) can do.

➤ Low cohesion focuses on what actions it involves, not what it does and what it contains.

➤ High cohesion focuses on what it should be doing,with only methods relating to the intention of the class.

➤ Coupling, refers to how related or dependent two classes/modules are toward each other.

➤ For low coupled changing something will not affect the other. In other hand for high coupled, change will be majorly impact the other.

➤ Because of this, good software design has high cohesion and low coupling.

10 - Heap and Stack means and differences?

➤ Memory management in Java handled by itself. The JVM divides the memory into two parts stack memory and heap memory.

➤ Stack is used to store the order of method execution and local variables while the heap memory stores the objects and it uses dynamic memory allocation and deallocation.

➤ Stack stores the variables, references to objects, and partial results.

➤ The stack memory is a physical space (in RAM) allocated to each thread at run time.

➤ Stack follows LIFO (Last-In-First-Out) order.

➤ Heap memory stores objects and JRE classes.

➤ Heap does not follow any order like the stack.

➤ Managing the memory automatically, by garbage collector that deletes the objects which are no longer being used.

➤ The elements are globally accessible.

11 – Exception means? Type of Exceptions?

➤ An event that occurs during the execution of a program that disrupts the normal flow of instructions is called an exception.

➤ Exceptions that are already available in Java libraries are referred to as built-in exception.

➤ There is also user-defined exception which in that case we want the program throw that exception.

➤ SQLException → Occurs while executing queries on a database related to the SQL syntax.

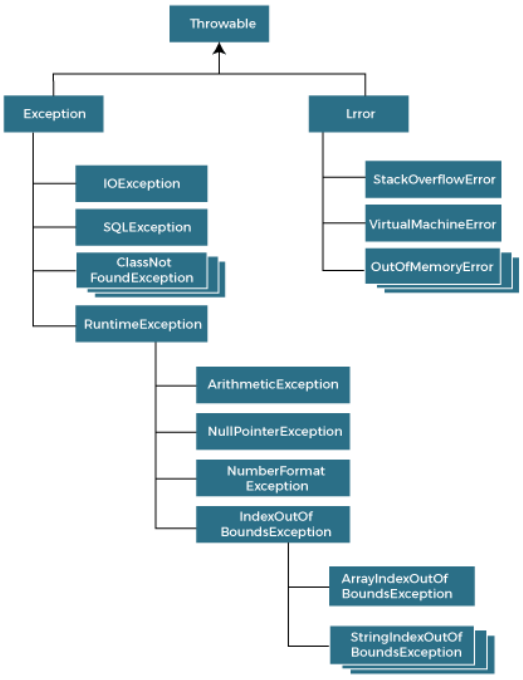
➤ IOException → Occurs while using file Input-Output stream operations.

➤ ArrayIndexOutofBound → Occurs when you try to access an array with an invalid index value.

➤ IllegalArgumentException → Occurs whenever an inappropriate or incorrect argument is passed to a method.

➤ NullPointerException → Occurs when you try to access an object with the help of a reference variable whose current value is null or empty.

➤ This is the java exception hierarchy.



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

➤ Clean code is code that is easy to understand and easy to change.

➤ Code is easy to extend and refactor, and it’s easy to fix bugs in the codebase.

13 - What is the method of hiding in Java?

➤ If a subclass defines a static method with the same signature as a static method in the super class in such a case, the method in the subclass hides the one in the superclass.

➤ The mechanism is known as method hiding. It happens because static methods are resolved at compile time and can't be overridden.

➤ Difference between the method overriding is methods are static, not instance.

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

➤ Abstraction hides the unnecessary detail but shows the essential information.

➤ Abstraction focuses on the external lookout while encapsulation focuses on internal working.

➤ Encapsulation hides the code and data into a single entity or unit.

➤ The objects are encapsulated that helps to perform abstraction.