**Week 3 Assesment**

1. **What is Java Virtual Machine (JVM) ?**

* JVM is the runtime virtual machine in which java program is run.
* It creates extra layers to the hardware.
* It takes the compile byte code and makes a specific code and runs it.
* Different implementations are available based on their choices.
* It plays most essential part making java platform agnostic.

1. **How aggregation and composition are different concepts?**

|  |  |
| --- | --- |
| Aggregation | Composition |
| It represents “has a” relationship where objects can relate, but their lifecycle is independent. | it represents “whole-part” relationship, their lifecycle is tightly related. |
| Objects can exit independently | Here child object are exclusiove complonents of parent object |
| The responsibility for creating and deleting objects lies with the individual objects. | The composite object is responsible for creating and deleting its components. |

1. **How will you implement method overloading in Java?**

* Method overloading in java helps to allow a class having multiple methods with same name but different parameters.
* It could be the same number of parameters but different types or both.
* When a method is called, the Java compiler determines the appropriate method to execute based on the method signature.

1. **What is the difference between method overloading and method overriding in Java?**

|  |  |
| --- | --- |
| **Method Overloading** | **Method Overriding** |
| Allows a class to have multiple methods with the same name but different parameter lists. | Involves creating a method in the subclass with the same signature as a method in its superclass. |
| Distinction based on number or types of parameters. | Distinction based on the method signature (name, return type, and parameter types). |
| Provides compile-time polymorphism | Provides runtime polymorphism. |
| Occurs within the same class. | Occurs between a superclass and its subclass |

1. **What is Abstraction in Object Oriented programming?**

* Abstraction is design principle of separating the interface from the implementation so that the consumners or clients only concerned about the interface.
* Ex: buttons on the electronic devices vs internal circutes
* Manifest the class design interface.(some time abstract class)

1. **What is Serialization and Deserialization conceptually?**
   1. **Serialization :** It is a process saving/storing the object to the hard drive (file). s like packing an object into a box, converting it into a format that can be easily stored or sent over the internet
   2. **Deserialization :** It's the process of restoring a serialized object to its complex, in-memory representation. **I**tis like unpacking the object from the box, retrieving it from the stored or transmitted format (such as a file or over the internet), and reconstructing it back to its original, usable form.
2. **What is Garbage Collection in Java?**

Process of removing unused and orphaned objects from heap is called garbage collections. This helps prevent memory clutter, making your program run more smoothly and avoiding potential issues.

1. **Why do we use finalize() method in Java?**

* Final key word makes something in final state and not to be changed.
* If we declare final key word to the variable it makes them constant
* If we declere final on objecrt reference, the reference is constatnt, but the object instance is not.
* For Class : not extandable (can not be inherited)
* For Methods: non overridable (can only be in herited)

1. **What are the advantages of Multithreading?**

* Allows parallel execution of threads, making better use of multi-core processors and potentially speeding up the overall performance of the program.
* Enables multiple tasks to run concurrently, improving responsiveness, especially in scenarios where waiting for certain operations is common.
* Two types:

Thread based (thread based multi tasking.)

Process based (Complete base application which means different apps does task same time)

1. **What are the different types of Dependency Injection in Spring Framework?**

There are different types are

**Constructor Injection:**

* + - Dependencies are injected through the constructor.
    - This is the most common type of Dependency Injection in Spring.
    - Dependencies are provided when creating an instance of the class.

**Setter Injection:**

* + - Dependencies are injected through setter methods.
    - The container uses setter methods to inject dependencies after creating an instance.
    - Provides flexibility for optional dependencies.

**Method Injection:**

* + - Dependencies are injected through methods.
    - Annotated methods (e.g., with @Autowired) are used for injection.
    - Dependencies are injected when the method is invoked**.**