Session-5 Interview Theory Questions

1.What is the purpose of a default constructor in Java?
A default constructor is a constructor that is provided by Java if no other constructors are defined in a class. Its purpose is to initialize the object's state with default values, such as zero or null. If a class does not have any constructors defined, Java will automatically provide a default constructor for that class. If a class has at least one constructor defined, but does not have a default constructor, then it cannot be instantiated without passing arguments to one of its constructors.

2**. What is the difference between a constructor and a static initializer in Java?**
A constructor is used to initialize the state of an object when it is created, while a static initializer is used to initialize the state of a class when it is loaded into memory. A constructor is called every time an object is created, while a static initializer is called only once, when the class is loaded into memory for the first time. Additionally, a constructor can have parameters, while a static initializer cannot.
3. **Can a constructor call another constructor in the same class?**
Yes, a constructor can call another constructor in the same class using this keyword. This is called constructor chaining. When one constructor calls another constructor using this, it must be the first statement in the constructor body. This allows the constructor being called to complete its initialization before the calling constructor continues its own initialization.
4. **What is the difference between constructor overloading and method overloading in Java?**

Constructor overloading is the practice of defining multiple constructors in a class with different parameter lists. This allows objects to be created in different ways depending on the arguments passed to the constructor. Method overloading, on the other hand, is the practice of defining multiple methods in a class with the same name but different parameter lists. This allows different methods to perform different operations on the same object, depending on the arguments passed to the method. The main difference between constructor overloading and method overloading is that constructors are used for object initialization, while methods are used for performing operations on objects.

5. **What does the static keyword mean in Java?**
The static keyword is used to declare a variable, method, or block of code that belongs to the class itself, rather than to any individual object of the class. A static variable or method can be accessed without creating an instance of the class, which means that it can be used by all instances of the class.
6. **What is the difference between a static method and a non-static method in Java?**

A static method is a method that belongs to the class itself, rather than to any individual object of the class. It can be accessed without creating an instance of the class and cannot access non-static variables or methods directly. A non-static method, on the other hand, is a method that belongs to an individual object of the class and can access both static and non-static variables and methods of that object.

7. **Can a static method access a non-static variable in Java?**
No, a static method cannot access a non-static variable in Java directly. To access a non-static variable from a static method, the variable must be declared as static or an object of the class must be created first.

8. **What is a static block in Java?**

A static block is a block of code that is executed when a class is loaded into memory for the first time. It is defined using the static keyword followed by a pair of curly braces. static blocks can be used to perform initialization of static variables, or to execute any other code that should be run once when the class is loaded.

9. **What is the purpose of the static final keyword in Java?**
The static final keyword is used to declare a constant in Java. A static final variable is a variable that belongs to the class itself, cannot be changed after it is initialized, and can be accessed without creating an instance of the class. It is conventionally named in all uppercase letters with underscores separating words (e.g. MY_CONSTANT).

10. **What is coupling and cohesion in java?**
Coupling and cohesion are two important concepts in software engineering that are used to measure the quality of software design. In Java, coupling refers to the degree of interdependence between classes or modules, while cohesion refers to the degree to which the responsibilities of a single module or class are related.
Coupling can be classified into several types, including content coupling, common coupling, control coupling, stamp coupling, and data coupling. Content coupling occurs when one class modifies the internal state of another class directly. Common coupling happens when two or more classes share the same global data. Control coupling occurs when one class has knowledge of the control flow of another class. Stamp coupling happens when data is passed between classes in a large data structure. Data coupling happens when two classes share only simple data types or arguments.
On the other hand, cohesion is classified into functional cohesion, sequential cohesion, communicational cohesion, procedural cohesion, temporal cohesion, and logical cohesion. Functional cohesion occurs when all the functions in a module or class are related to a single task or function. Sequential cohesion occurs when the output of one

function is used as input to another function. Communicational cohesion occurs when functions operate on the same data. Procedural cohesion occurs when functions are grouped together based on the order they are called. Temporal cohesion occurs when functions are grouped together based on time. Logical cohesion occurs when functions are grouped together based on the logical relationship between them.
In general, high cohesion and low coupling are desirable in software design as they lead to modular, maintainable, and flexible code that can be easily tested and modified.