Cognizant Full Stack Jr. Java Developer Apprentice QUESTION SET:

DPI
DISCOVERY PARTNERS INSTITUTE
PART OF THE UNIVERSITY OF ILLINOIS SYSTEM

1.) What is Object Oriented Paradigm?

The goal of an object oriented paradigm is to represent the real world while writing code. It does not focus on the problem that needs to be solved but focuses on the objects that make up the system.

Some Features:

- OOP breaks a problem into a number of entities called objects and then builds data and functions around them.
- It treats data as a critical element in the program development and therefore restricts the flow of data.
- OOP protects the data from accidental modification from outside functions.
- Objects of the different classes can interact easily through functions.
- The object-oriented paradigm follows a bottom-up approach.

2.) What are the main features of OOPs?

OOPs or Object-Oriented Programming mainly comprises of the below four features, and make sure you don't miss any of these:

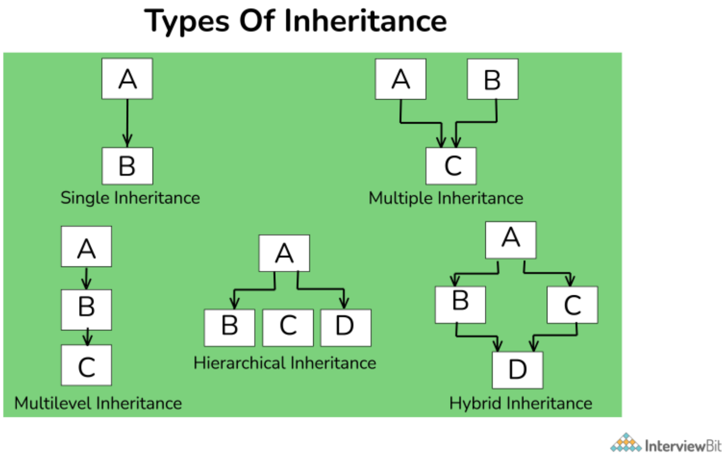
- Inheritance
- Encapsulation
- Polymorphism
- Abstraction

3.) What is inheritance?

Inheritance in Java is a mechanism in which one object acquires all the properties and behaviors of a parent object. The idea behind inheritance in Java is that you can create new classes that are built upon existing classes. When you inherit from an existing class, you can reuse methods and fields of the parent class. Moreover, you can add new methods and fields in your current class also.

Syntax:

```
class Subclass-name extends Superclass-name
{
    //methods and fields
}
```



4.) What is encapsulation?

One can visualize Encapsulation as the method of putting everything that is required to do the job, inside a capsule and presenting that capsule to the user. What it means is that by Encapsulation, all the necessary data and methods are bound together and all the unnecessary details are hidden to the normal user. So encapsulation is the process of binding data members and methods of a program together to do a specific job, without revealing unnecessary details.

5.) What is polymorphism?

Polymorphism refers to the process by which some code, data, method, or object behaves differently under different circumstances or contexts. Compile-time polymorphism and Run time polymorphism are the two types of polymorphisms in OOPs languages.

6.) What is compile-time polymorphism?

Compile time polymorphism, also known as Static Polymorphism, refers to the type of Polymorphism that happens at compile time. What it means is that the compiler decides what shape or value has to be taken by the entity in the picture.

7.) What is run-time polymorphism?

Runtime polymorphism, also known as Dynamic Polymorphism, refers to the type of Polymorphism that happens at the run time. What it means is it can't be decided by the compiler. Therefore, what shape or value has to be taken depends upon the execution.

8.) What is Abstraction?

If you are a user, and you have a problem statement, you don't want to know how the components of the software work, or how it's made. You only want to know how the software solves your problem. Abstraction is the method of hiding unnecessary details from the necessary ones. It is one of the main features of OOPs.

For example, consider a car. You only need to know how to run a car, and not how the wires are connected inside it. This is obtained using Abstraction.

9.) Why is Java considered dynamic?

It is designed to adapt to an evolving environment. Java programs can carry extensive amount of run-time information that can be used to verify and resolve accesses to objects on run-time.

10.) What is Java Virtual Machine and how it is considered in the context of Java's platform independent feature?

When Java is compiled, it is not compiled into platform specific machine, rather into platform independent byte code. This byte code is distributed over the web and interpreted by virtual Machine (JVM) on whichever platform it is being run.

11.) What is a constructor?

Constructor gets invoked when a new object is created. Every class has a constructor. If we do not explicitly write a constructor for a class the java compiler builds a default constructor for that class.

12.) Define Destructor?

A destructor is a method which is automatically called when the object is made of scope or destroyed. Destructor name is also same as class name but with the tilde symbol before the name.

13.) What is an Inline function?

An inline function is a technique used by the compilers and instructs to insert complete body of the function wherever that function is used in the program source code.

14.) What is a virtual function?

A virtual function is a member function of a class, and its functionality can be overridden in its derived class. This function can be implemented by using a keyword called virtual, and it can be given during function declaration.
A virtual function can be declared using a token(virtual) in C++. It can be achieved in C/Python Language by using function pointers or pointers to function.

15.) What is a friend function?
A friend function is a friend of a class that is allowed to access to Public, private, or protected data in that same class. If the function is defined outside the class cannot access such information.
A friend can be declared anywhere in the class declaration, and it cannot be affected by access control keywords like private, public, or protected.

16.) What is function overloading?
Function overloading is a regular function, but it can perform different tasks. It allows the creation of several methods with the same name which differ from each other by the type of input and output of the function.

17.) What is operator overloading?
Operator overloading is a function where different operators are applied and depends on the arguments. Operator,-,* can be used to pass through the function, and it has its own precedence to execute

18.) What is an abstract class?
An abstract class is a class which cannot be instantiated. Creation of an object is not possible with an abstract class, but it can be inherited. An abstract class can contain only an Abstract method. Java allows only abstract method in abstract class while other languages allow non- abstract method as well.

19.) What is a ternary operator?
The ternary operator is said to be an operator which takes three arguments. Arguments and results are of different data types, and it depends on the function. The ternary operator is also called a conditional operator.

20.) What is the use of finalize method?
Finalize method helps to perform cleanup operations on the resources which are not currently used. Finalize method is protected, and it is accessible only through this class or by a derived class.

21.) What are the different types of arguments?

A parameter is a variable used during the declaration of the function or subroutine, and arguments are passed to the function body, and it should match with the parameter defined. There are two types of Arguments.

- Call by Value – Value passed will get modified only inside the function, and it returns the same value whatever it is passed into the function.
- Call by Reference – Value passed will get modified in both inside and outside the functions and it returns the same or different value.

22.) What is the super keyword?

The super keyword is used to invoke the overridden method, which overrides one of its superclass methods. This keyword allows to access overridden methods and also to access hidden members of the superclass.

It also forwards a call from a constructor to a constructor in the superclass.

23.) What is method overriding?

Method overriding is a feature that allows a subclass to provide the implementation of a method that overrides in the main class. It will override the implementation in the superclass by providing the same method name, same parameter, and same return type.

24.) What is an interface?

An interface is a collection of an abstract method. If the class implements an interface, it thereby inherits all the abstract methods of an interface.

Java uses Interface to implement multiple inheritances.

25.) What is exception handling?

An exception is an event that occurs during the execution of a program. Exceptions can be of any type – Runtime exception, Error exceptions. Those exceptions are adequately handled through exception handling mechanism like try, catch, and throw keywords.

26.) What are tokens?

A compiler recognizes a token, and it cannot be broken down into component elements. Keywords, identifiers, constants, string literals, and operators are examples of tokens.

Even punctuation characters are also considered as tokens. Example: Brackets, Commas, Braces, and Parentheses.

27.) What is the main difference between overloading and overriding?

Overloading is static Binding, whereas Overriding is dynamic Binding.
Overloading is nothing but the same method with different arguments, and it may or may not return the equal value in the same class itself.
Overriding is the same method names with the same arguments and return types associated with the class and its child class.

28.) What is the difference between arrays and collections?
An array in Java is a group of like-typed variables referred to by a common name.
A group of individual objects which are represented as a single unit is known as the collection of the objects. In Java, a separate framework named the "Collection Framework" has been defined in JDK 1.2 which holds all the collection classes and interface in it.

29.) What is a database?
The database is a system used to save and process data in an efficient manner. The way it stores information is usually in a well-organized structure, and we can easily manipulate this data according to our requirements.

30.) What is SQL?
SQL or Structured Query Language is basically the language that we (the user) use to communicate with the Databases and get our required interpretation of data out of it. It is used for storing, manipulating and retrieving data out of a database.

31.) What are CRUD operations in SQL?
CRUD is an abbreviation for Create, Read, Update and Delete. These 4 operations comprise the most basic database operations. The relevant commands for these 4 operations in SQL are:

- Create: INSERT
  ```
  INSERT INTO name_of_table(column1, column2, ....)
    VALUES(value1, value2, ....)
  ```

- Read: SELECT
  ```
  SELECT column1,column2,.. FROM name_of_table;
  ```

- Update: UPDATE
  ```
  UPDATE name_of_table
  SET column1=value1,column2=value2,...
  WHERE conditions...;
  ```

- Delete: DELETE (deletes contents of the table but not the table itself)

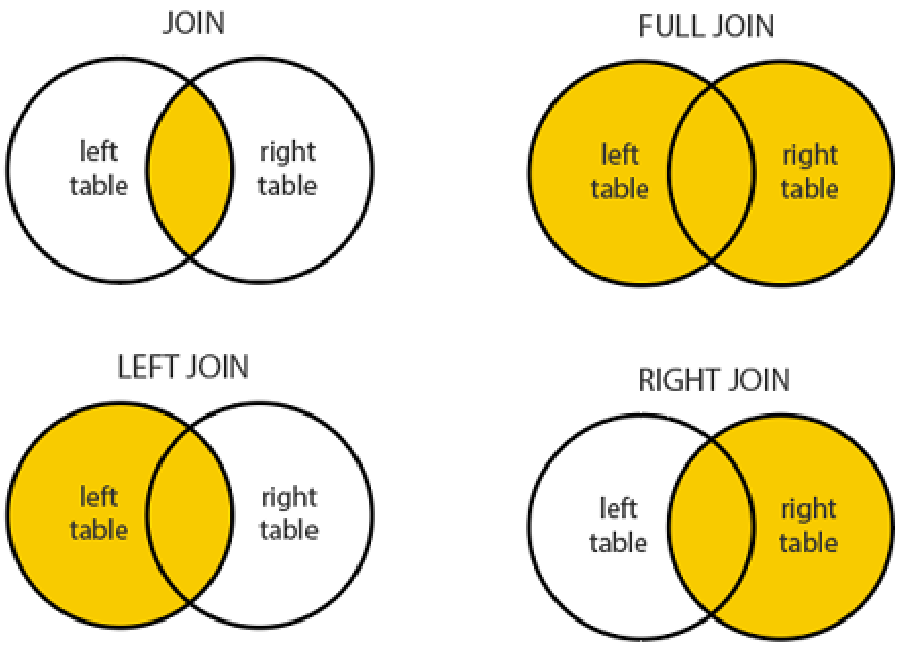
  ```
  DELETE FROM name_of_table
  WHERE condition1, condition2, ...;
  ```

  DROP: (drops content along with the table)

  ```
  DROP TABLE <table_name>;
  ```

32.) What is join?

A JOIN clause is used to combine rows from two or more tables, based on a related column between them. The types are:



Syntax:

```
SELECT columns
FROM lefttable
INNER JOIN righttable
ON lefttable.column = righttable.column;
```