SCREENING INTERVIEW QUESTIONS

## CORE JAVA

1. List the OO concepts and features and describe each in short.

Class, Instance, Inheritance, Abstraction, Encapsulation, Polymorphism. An overview answer with a small example will suffice. <https://www.javatpoint.com/java-oops-concepts>

A: OOPs (Object Oriented Programming System)

Java oops concepts Object mean a real word entity such as pen, chair, table etc. Object-Oriented Programming is a methodology or paradigm to design a program using classes and objects. It simplifies the software development and maintenance by providing some concepts:

Object

Class

Inheritance

Polymorphism

Abstraction

Encapsulation

Object

Any entity that has state and behavior is known as an object. For example: chair, pen, table, keyboard, bike etc. It can be physical and logical.

Class

Collection of objects is called class. It is a logical entity.

Inheritance

When one object acquires all the properties and behaviours of parent object i.e. known as inheritance. It provides code reusability. It is used to achieve runtime polymorphism.

polymorphism in java oops concepts

Polymorphism

When one task is performed by different ways i.e. known as polymorphism. For example: to convince the customer differently, to draw something e.g. shape or rectangle etc.

In java, we use method overloading and method overriding to achieve polymorphism.

Another example can be to speak something e.g. cat speaks meaw, dog barks woof etc.

Abstraction

Hiding internal details and showing functionality is known as abstraction. For example: phone call, we don't know the internal processing.

In java, we use abstract class and interface to achieve abstraction.

encapsulation in java oops concepts

Encapsulation

Binding (or wrapping) code and data together into a single unit is known as encapsulation. For example: capsule, it is wrapped with different medicines.

A java class is the example of encapsulation. Java bean is the fully encapsulated class because all the data members are private here.

1. What is Overloading vs. Overriding in polymorphism? Differences with a small code example as in <https://www.javatpoint.com/method-overloading-vs-method-overriding-in-java>

A:

|  |  |  |
| --- | --- | --- |
| **No.** | **Method Overloading** | **Method Overriding** |
| 1) | Method overloading is used *to increase the readability* of the program. | Method overriding is used *to provide the specific implementation* of the method that is already provided by its super class. |
| 2) | Method overloading is performed *within class*. | Method overriding occurs *in two classes* that have IS-A (inheritance) relationship. |
| 3) | In case of method overloading, *parameter must be different*. | In case of method overriding, *parameter must be same*. |
| 4) | Method overloading is the example of *compile time polymorphism*. | Method overriding is the example of *run time polymorphism*. |
| 5) | In java, method overloading can't be performed by changing return type of the method only. *Return type can be same or different* in method overloading. But you must have to change the parameter. | *Return type must be same or covariant* in method overriding. |

Java Method Overloading example

class OverloadingExample{

static int add(int a,int b){return a+b;}

static int add(int a,int b,int c){return a+b+c;}

}

Java Method Overriding example

class Animal{

void eat(){System.out.println("eating...");}

}

class Dog extends Animal{

void eat(){System.out.println("eating bread...");}

}

1. What are covariant returns?

The covariant return type specifies that the return type may vary in the same direction as the subclass.

Before Java5, it was not possible to override any method by changing the return type. But now, since Java5, it is possible to override method by changing the return type if subclass overrides any method whose return type is Non-Primitive but it changes its return type to subclass type. Let's take a simple example:

1. Instance vs. Object vs Class. (<https://alfredjava.wordpress.com/2008/07/08/class-vs-object-vs-instance/>)

In short, An object is a software bundle of related state and behavior. A class is a blueprint or prototype from which objects are created. An instance is a single and unique unit of a class.

Example, we have a blueprint (class) represents student (object) with fields like name, age, course (class member). And we have 2 students here, Foo and Bob. So, Foo and Bob is 2 different instances of the class (Student class) that represent object (Student people).

Object

Real world objects shares 2 main characteristics, state and behavior. Human have state (name, age) and behavior (running, sleeping). Car have state (current speed, current gear) and state (applying brake, changing gear). Software objects are conceptually similar to real-world objects: they too consist of state and related behavior. An object stores its state in fields and exposes its behavior through methods.

Class

Class is a “template” / “blueprint” that is used to create objects. Basically, a class will consists of field, static field, method, static method and constructor. Field is used to hold the state of the class (eg: name of Student object). Method is used to represent the behavior of the class (eg: how a Student object going to stand-up). Constructor is used to create a new Instance of the Class.

Instance

An instance is a unique copy of a Class that representing an Object. When a new instance of a class is created, the JVM will allocate a room of memory for that class instance.

1. What are constructors in java and what are its types? <https://www.javatpoint.com/java-constructor>
   1. Can constructors be private?
   2. What is a copy constructor? Is it available in Java, how do you copy one object to another (A: Using Cloning, using constructor. See section ‘Java Copy Constructor’ in the above link)
2. What is deep cloning and shallow cloning <http://javaconceptoftheday.com/difference-between-shallow-copy-vs-deep-copy-in-java/>
3. Which version of Java is the latest? (Java 9).

What are some of the new features in Java8?

<https://www.javatpoint.com/java-8-features>

Candidate should know at least these 3 out of these: Labmda Expressions, Static methods in interfaces, default methods, foreach() method. If they can provide a brief explanation of each, it would be a bonus.

1. Explain the exceptions types in Java. <https://www.javatpoint.com/exception-handling-in-java>. Candidate needs to be able to provide the Hierarchy diagram of the Exceptions in Java and at least 2 different types under each of ‘Exception’ and ‘Error’.
2. What are checked and unchecked exceptions? <https://www.javatpoint.com/exception-handling-in-java>. See section of Checked and Unchecked exceptions. Candidate needs to be able to provide brief explanation of both and provide 2 exceptions names under each of checked and unchecked exceptions.
3. What is the difference between interface and abstract class? <https://www.javatpoint.com/difference-between-abstract-class-and-interface> candidate needs to provide at least 3 differences
4. What is the collections framework in Java? <https://www.javatpoint.com/collections-in-java>. Candidate needs to show the hierarchy as in the above link and provide at least 2 types with explanation under ‘List’ and 1 under ‘Queue’ and 2 under ‘Set’
5. What is the Map interface in Java? <https://www.javatpoint.com/java-map>
6. What is the difference between hashmap and hashtable? <https://www.javatpoint.com/difference-between-hashmap-and-hashtable> 3 differences needed.
7. What is the difference between Arraylist and LinkedList <https://www.javatpoint.com/difference-between-arraylist-and-linkedlist> candidate needs to provide at least 2 differences?
8. What does the hashcode methods do? <https://www.javaworld.com/article/2074996/hashcode-and-equals-method-in-java-object---a-pragmatic-concept.html> If candidate can provide brief explanation as in the 2 paragraphs for hashcode and equals in the above example it would suffice.
9. Explain the lifecycle of a thread with the different states. <https://www.javatpoint.com/life-cycle-of-a-thread>
10. What is a deadlock? <https://www.javatpoint.com/deadlock-in-java>
11. What is an inner class in Java? What is the difference between inner class and nested class? <https://www.javatpoint.com/java-inner-class>
12. What are the different access modifiers in java? <https://www.geeksforgeeks.org/access-modifiers-java/> Provide a brief explanation of each. Need to mention all 4.
13. What are the non-access modifiers in Java? <https://www.tutorialspoint.com/java/java_nonaccess_modifiers.htm> candidate needs to mention at least 3.

## J2EE (JSP, SERVLETS)

1. What are the implicit objects in JSP? <https://www.tutorialspoint.com/jsp/jsp_implicit_objects.htm> Need to name and give a brief description of at least 5
2. Standard actions and directives. <https://www.tutorialspoint.com/jsp/jsp_actions.htm> Need to name and give a brief description of at least 4.
3. Explain the servlet lifecycle. <https://www.tutorialspoint.com/servlets/servlets-life-cycle.htm> candidate needs to explain the init(), service() and destroy() methods briefly
4. Difference between variable declared in “declaration” part and “scriptlet” part. i.e :

<%! int i=0;%> and <%int i=0%>

A: <http://www.xyzws.com/jspfaq/what-is-the-difference-between-variable-declared-inside-a-declaration-part-and-variable-declared-in-scriplet-part/12>

1. What is the difference between Jsp forward vs response.sendRedirect(url)

<http://javarevisited.blogspot.in/2011/09/sendredirect-forward-jsp-servlet.html> Need at least 2 differences from the table.

### JAVASCRIPT/HTML/CSS

1. Difference between visibility:hidden and display:none.

<https://stackoverflow.com/questions/133051/what-is-the-difference-between-visibilityhidden-and-displaynone>

1. What are the values for position CSS

A: absolute, static,relative,fixed,inherit

1. What is the difference between == and === in JS. (Answer no 13 in <https://www.javatpoint.com/javascript-interview-questions>)
2. What is the difference between null and undefined (Answer 24 in <https://www.javatpoint.com/javascript-interview-questions> )
3. Ask for output of any 1 of the 3 of the coding question 2 in link below. rotate between candidates. <https://www.codementor.io/nihantanu/21-essential-javascript-tech-interview-practice-questions-answers-du107p62z>
4. What does the ‘use strict’ directive do in Javascript. <https://www.toptal.com/javascript/interview-questions>. See section ‘https://www.w3schools.com/js/js\_strict.asp’ for answer.
5. What is ‘hoisting’ in Javascript? <https://www.w3schools.com/js/js_hoisting.asp>
6. What is JSON? Provide an example. <https://www.w3schools.com/js/js_json.asp>
7. What are the different scopes in Javascript? <https://www.w3schools.com/js/js_scope.asp>
8. What is an Event? Provide an example. List some common html events (candidate should be able to list at least 4) <https://www.w3schools.com/js/js_events.asp>

### SPRING/HIBERNATE

1. What is spring?

A. Spring is an open source development framework for Enterprise Java. The core features of the Spring Framework can be used in developing any Java application, but there are extensions for building web applications on top of the Java EE platform. Spring framework targets to make Java EE development easier to use and promote good programming practice by enabling a POJO-based programming model.

2. What are features of spring?

A.

* Lightweight
* Inversion of control (IOC)
* Aspect oriented (AOP)
* Container
* MVC Framework
* JDBC Exception Handling
* Transaction Management

3. What is Spring Java Based Configuration?

A. Java based configuration option enables the user to write most of their spring configuration without XML but with the help of few Java-based annotations. For example:

Annotation @Configuration indicates that the class can be used by the Spring IoC container as a source of bean definitions. The @Bean annotation tells spring that a method annotated with @Bean will return an object that should be registered as a bean in the spring application context.

4. Explain spring architecture?

A.



5. Which are the spring framework modules?

A. The basic modules of the spring framework are:

* Core module
* Bean module
* Context module
* Expression Language module
* JDBC module
* ORM module
* OXM module
* Java Messaging Service(JMS) module
* Transaction module
* Web module
* Web-Servlet module
* Web-Struts module
* Web-Portlet module

6. Describe some of the standard spring events?

A. Spring provides the following standard events

* ContextRefreshedEvent
* ContextStartedEvent
* ContextStoppedEvent
* ContextClosedEvent
* RequestHandledEvent

7. What is Hibernate?

A. Hibernate is an Object-Relational Mapping(ORM) solution for JAVA and it raised as an open source persistent framework created by Gavin King in 2001. It is a powerful, high performance Object-Relational Persistence and Query service for any Java Application.

Hibernate maps Java classes to database tables and from Java data types to SQL data types and relieve the developer from 95% of common data persistence related programming tasks.

8. What are the advantages of using Hibernate?

A. Following are the advantages of using Hibernate.

* Hibernate takes care of mapping Java classes to database tables using XML files and without writing any line of code.
* Provides simple APIs for storing and retrieving Java objects directly to and from the database.
* If there is change in Database or in any table then the only need to change XML file properties.
* Abstract away the unfamiliar SQL types and provide us to work around familiar Java Objects.
* Hibernate does not require an application server to operate.
* Manipulates Complex associations of objects of your database.
* Minimize database access with smart fetching strategies.
* Provides Simple querying of data.

9. What is Session in hibernate?

A. Session is used to get a physical connection with a database. The Session object is lightweight and designed to be instantiated each time an interaction is needed with the database. Persistent objects are saved and retrieved through a Session object.

The session objects should not be kept open for a long time because they are not usually thread safe and they should be created and destroyed them as needed.

10. What are persistent classes in hibernate?

A. Java classes whose objects or instances will be stored in database tables are called

persistent classes in Hibernate.

11. What is root node of hbm.xml?

A. The mapping document is an XML document having <hibernate-mapping> as the root

element which contains all the <class> elements.

12. Is Session a thread-safe object?

A. No, Session is not thread-safe.

13. What is lazy loading?

A. Lazy loading is a technique in which objects are loaded on demand basis. Since Hibernate

3, lazy loading is by default, enabled so that child objects are not loaded when parent is

loaded.

14. What is hibernate configuration file?

A. Hibernate configuration file contains database specific configurations and used to initialize

SessionFactory. We provide database credentials or JNDI resource information in the

hibernate configuration xml file. Some other important parts of hibernate configuration

file is Dialect information, so that hibernate knows the database type and mapping file or

class details.

15. What are the collection types in Hibernate?

A. There are five collection types in hibernate used for one-to-many relationship mappings.

* Bag
* Set
* List
* Array
* Map

### SQL

1. What are the Advantages of SQL ?

A.

* SQL is easy to learn
* SQL is not a proprietary language can be used in ORACLE, SQL ,MYSQL etc.
* Complex and sophisticated database operations

2. When are we going to use truncate and delete?

A.

* TRUNCATE is a DDL command, whereas DELETE is a DML command.
* TRUNCATE is quicker than DELETE
* If a table is referenced by any foreign key constraints, then TRUNCATE won’t work.

3. What are the different types of SQL's statements?

A.

* DDL – Data Definition Language

For example, Create, Alter, Drop and Truncate table.

* DML – Data Manipulation Language

For example, insert Delete, Update and retrieving the data from the table.

* DCL – Data Control Language

For example, granting database access and set privileges to create tables

4. What is a trigger?

A. Database triggers are sets of commands that get executed when an event (Before Insert,

After Insert, On Update, On delete of a row) occurs on a table, views.

5. What are wild cards used in database for Pattern Matching?

A. SQL Like operator is used for pattern matching. SQL 'Like' command takes more time to

process. So before using "like" operator, consider suggestions given below on when and

where to use wild card search.

6. Define Join and explain different type of joins?

A. Join keyword is used to fetch data from related tables. "Join" returns rows when there is at

least one match in both tables. Types of joins are

Right Join

Return all rows from the right table, even if there are no matches in the left table.

Outer Join

Left Join

Return all rows from the left table, even if there are no matches in the right table.

Full Join

Return rows when there is a match in one of the tables

7. What are the advantages of DBMS?

* Redundancy is controlled.
* Unauthorised access is restricted.
* Providing multiple user interfaces.
* Enforcing integrity constraints.
* Providing backup and recovery.

8. What is normalization?

A. It is a process of analysing the given relation schemas based on their Functional Dependencies (FDs) and primary key to achieve the properties

* Minimizing redundancy
* Minimizing insertion, deletion and update anomalies.

9. What is indexing and what are the different kinds of indexing?

A. Indexing is a technique for determining how quickly specific data can be found. Types

* Binary search style indexing
* B-Tree indexing
* Inverted list indexing
* Memory resident table
* Table indexing

10. Define SQL Insert, Update, and Delete Statement?

A. INSERT INTO tableName(field1,field2,...)VALUES(value1, value2,...>);

UPDATE tableName SET field1 = value1, field2 = value2 WHERE field = <{expr}>;

DELETE FROM tableName WHERE <{where\_expression}>;