Core Java Basic Interview Questions And Answers Pdf

1. What are the important features of Java 8 release?
Answer: Java 8 has been released in March 2014, so it's one of the hot topics in java interview questions. If you answer this question clearly, it will show that you like to keep yourself up-to-date with the latest technologies.
Java 8 has been one of the biggest releases after Java 5 annotations and generics.

Some of the important features of Java 8 are:

Interface changes with the default and static methods
Functional interfaces and Lambda Expressions
Java Stream API for collection classes
Java Date Time API
I strongly recommend to go through the above links to get a proper understanding of each one of them, also read Java 8 Features.

2. Name some OOPS Concepts in Java?
Answer: Java is based on Object-Oriented Programming Concepts, the following are some of the OOPS concepts implemented in java programming.

- Abstraction
- Encapsulation
- Polymorphism
- Inheritance
- Association
- Aggregation
- Composition

3. What is the numeric promotion?
Answer: Numeric promotions of a numeric operator are used for the conversion of the operands into a common type. In order to perform calculation easily, numeric promotion, conversion is performed.
It is the conversion of a smaller numeric type to a larger numeric type so that integer and floating-point operations can be performed over it.
here byte, char, and short values are converted to int values.
The int values are converted to long values, and the long and float values are converted to double values.

4. What are the methods used to implement for the key Object in the Hash Map?
Answer: Equals and hash code methods are to be implemented In order to use any object as Key in Hash Map, in Java.

5. What is a JIT compiler?

Answer: Just-In-Time(JIT) compiler is used to improve performance. JIT compiles parts of the bytecode that has similar functionality which in turn reduces the amount of time needed for compilation. The term “compiler” here refers to a translator from the instruction set of a Java virtual machine (JVM) to the instruction set of a specific CPU.

6. Differentiate between a constructor and a method? Can we mark constructors final?
Answer: A constructor constructs the value, by providing data for the object. It is a special type of method that is used to initialize the object. The constructor has the same name as the class itself, has no return type, and is invoked using the new operator.
A method is an ordinary member function of a class. A method can be invoked using the dot operator and has its own name, and a return type.
No, declaring the constructor as final is not possible.

7. What is the final class?
Answer: A final class is a constant value of a final variable. Extending A final class is not possible ie., final class may not be sub-classed. A final method cannot be overridden when its class is inherited.

8. What is the multi-catch block in Java?
Answer: Multi-catch block makes the code shorter and cleaner when every catch block has a similar code. We can catch multiple exceptions in a single catch block using this feature.

9. What are the important features of Java 10 release?
Answer: Java 10 is the first every-six-months from Oracle corporation, so it’s not a major release like earlier versions. However, some of the important features of **Java 10 are:**
Local-Variable Type Inference
Enhance java.util.Locale and related APIs to implement additional Unicode extensions of BCP 47 language tags.
Enable the HotSpot VM to allocate the Java object heap on an alternative memory device, such as an NV-DIMM, specified by the user.
Provide a default set of root Certification Authority (CA) certificates in the JDK.
Java 10 is mostly a maintenance release, however, I really liked the local variable type inference feature.

10. What are the important features of Java 9 release?
Answer:

Public: When a class is public, the public class is visible in other packages, the field is visible everywhere.
Private: Private variables or methods can be used by an instance of the same class only which declares the variable or method. A private feature can be accessed by the class that owns the feature.

Protected: Protected variable is available to all classes in the same package. It is also available to all subclasses of the class that owns the protected feature. Subclasses that reside in a different package also is provided with access from the class that owns the protected feature.

11. State the significance of public, private, protected class?
Answer:

Public: When a class is public, the public class is visible in other packages, the field is visible everywhere.
Private: Private variables or methods can be used by an instance of the same class only which declares the variable or method. A private feature can be accessed by the class that owns the feature.
Protected: Protected variable is available to all classes in the same package. It is also available to all subclasses of the class that owns the protected feature. Subclasses that reside in a different package also is provided with access from the class that owns the protected feature.

12. What is JVM and is it platform independent?
Answer: Java Virtual Machine (JVM) is the heart of java programming language. JVM is responsible for converting byte code into machine-readable code. JVM is not platform independent, thats why you have different JVM for different operating systems. We can customize JVM with Java Options, such as allocating minimum and maximum memory to JVM. It's called virtual because it provides an interface that doesn't depend on the underlying OS.

13. What is the difference between JDK and JVM?
Answer: Java Development Kit (JDK) is for development purpose and JVM is a part of it to execute the java programs.
JDK provides all the tools, executables and binaries required to compile, debug and execute a Java Program. The execution part is handled by JVM to provide machine independence.

14. Differentiate between String Buffer and String Builder in Java?
Answer: The only difference between StringBuffer and StringBuilder is that StringBuffer methods are synchronized while StringBuilder is not synchronized. StringBuilder in Java was introduced in Java 5.

15. What is Enum in Java?
Answer: Enum was introduced in Java 1.5 as a new type whose fields consists of a fixed set of constants. For example, in Java, we can create Direction as an enum with fixed fields as EAST, WEST, NORTH, SOUTH.
enum is the keyword to create an enum type and similar to a class. Enum constants are implicitly static and final. Read more in detail at java enum.

16. What are Java Annotations?
Answer: Java Annotations provide information about the code and they have no direct effect on the code they annotate. Annotations are introduced in Java 5. Annotation is metadata about the

program embedded in the program itself. It can be parsed by the annotation parsing tool or by the compiler. We can also specify annotation availability to either compile-time only or till runtime also. Java Built-in annotations are @Override, @Deprecated and @SuppressWarnings. Read more at java annotations.

17. What is the Java Reflection API? Why it's so important to have?
Answer: Java Reflection API provides the ability to inspect and modify the runtime behavior of java application. We can inspect a java class, interface, enum and get their methods and field details. Reflection API is an advanced topic and we should avoid it in normal programming. Reflection API usage can break the design pattern such as Singleton pattern by invoking the private constructor i.e violating the rules of access modifiers.
Even though we don't use Reflection API in normal programming, it's very important to have. We can't have any frameworks such as Spring, Hibernate or servers such as Tomcat, JBoss without Reflection API. They invoke the appropriate methods and instantiate classes through reflection API and use it a lot for other processing.

18. What is method overloading and method overriding?
Answer:
Method Overloading:
In Method Overloading, Methods of the same class shares the same name but each method must have a different number of parameters or parameters having different types and order.
Method Overloading is to "add" or "extend" more to the method's behavior.
It is a compile-time polymorphism.
The methods must have a different signature.
It may or may not need inheritance in Method Overloading.
Let's take a look at the example below to understand it better. (Online Java Training Courses)

19. What is the difference between path and classpath variables?
Answer: PATH is an environment variable used by the operating system to locate the executables. That's why when we install Java or want any executable to be found by OS, we need to add the directory location in the PATH variable. If you work on Windows OS, read this post to learn how to set up the PATH variable on Windows.
Classpath is specific to Java and used by java executables to locate class files. We can provide the classpath location while running java application and it can be a directory, ZIP files, JAR files, etc.

20. What is the importance of main method in Java?
Answer: main() method is the entry point of any standalone java application. The syntax of main method is public static void main(String args[]).
main method is public and static so that java can access it without initializing the class. The input parameter is an array of String through which we can pass runtime arguments to the java program. Check this post to learn how to compile and run a java program.

21. What is the static keyword?
Answer: static keyword can be used with class-level variables to make it global i.e all the objects will share the same variable.
static keyword can be used with methods also. A static method can access only static variables of class and invoke only static methods of the class. (interview questions and answers)

22. What is finally and finalize in java?
Answer: finally block is used with try-catch to put the code that you want to get executed always, even if an exception is thrown by the try-catch block. finally block is mostly used to release resources created in the try block.
finalize() is a special method in Object class that we can override in our classes. This method gets called by the garbage collector when the object is getting garbage collected.

This method is usually overridden to release system resources when object is garbage collected.

23. What is an abstract class?
Answer: Abstract classes are used in java to create a class with some default method implementation for subclasses. An abstract class can have abstract method without body and it can have methods with implementation also.
abstract keyword is used to create an abstract class. Abstract classes can’t be instantiated and mostly used to provide a base for sub-classes to extend and implement the abstract methods and override or use the implemented methods in the abstract class. Read important points about abstract classes at java abstract class.

24. What is the difference between @Before and @BeforeClass annotation?
Answer: The code marker @Before is executed before each test, while @BeforeClass runs once before the entire test fixture. If your text class has ten tests, @Before code will be executed ten times, but @BeforeClass will be executed only once.
In general, you would use @BeforeClass when numerous tests are required to share the same computationally expensive setup code. Starting a database connection falls into this category. You are able to move code from @BeforeClass into @Before, but your test run may be delayed. @BeforeClass is run as a static initializer, therefore it will run before the class instance of your test fixture is created.

25. Is it possible to make an array volatile in Java?
Answer: Yes, it is possible to make an array volatile in Java, but only the reference which is pointing to an array, not the whole array. Therefore, if one thread changes the reference variable points to another array, which will provide a volatile guarantee.
However, if several threads are altering particular array elements, there won’t be any happens before assurance provided by the volatile modifier for such modification.
If the purpose is to provide memory visibility guarantee for individual indices of the array, volatile is of no practical use for you. Instead, you must rely on an alternative mechanism for thread-safety in that particular case, e.g. use of a synchronized keyword.

26. From the two, which would be easier to write: synchronization code for ten threads or two threads?
Answer: Both will have the same level of complexity regarding writing the code because synchronization is independent of the number of threads, although the choice of synchronization could be subject to the number of threads because this presents more conflict.
Therefore, you would opt for an advanced synchronization technique, e.g. lock stripping, which requires more intricate code and proficiency.

27. What's the difference between unit, integration and functional testing?
Answer: A unit tests a small isolated piece of code such as a method. It doesn't interact with any other systems such as a database or another application.
An integration test tests how your code plays with other systems, such as a database, a cache or a third-party application.
A functional test is the testing of the complete functionality of an application. This may involve the use of an automated tool to carry out more complex user interactions with your system. It tests certain flows/use cases of your application.

28. What are wrapper classes?
Answer: A unit tests a small isolated piece of code such as a method. It doesn't interact with any other systems such as a database or Wrapper classes converts the java primitives into the reference types (objects). Every primitive data type has a class dedicated to it. These are known as wrapper classes because they "wrap" the primitive data type into an object of that class. Refer to the below image which displays different primitive type, wrapper class, and constructor argument.

29. What is the difference between equals() and?
Answer: Equals() method is defined in Object class in Java and used for checking equality of two objects defined by business logic.
"==" or equality operator in Java is a binary operator provided by Java programming language and used to compare primitives and objects. public boolean equals(Object o) is the method provided by the Object class. The default implementation uses == operator to compare two objects. For example, method can be overridden like String class. equals() method is used to compare the values of two objects.

30. What is the difference between abstract classes and interfaces?
Answer:
Abstract Class Interfaces
An abstract class can provide complete, default code and/or just the details that have to be overridden. An interface cannot provide any code at all, just the signature.
In the case of an abstract class, a class may extend only one abstract class. A Class may implement several interfaces.
An abstract class can have non-abstract methods. All methods of an Interface are abstract.
An abstract class can have instance variables. An Interface cannot have instance variables
An abstract class can have any visibility: public, private, protected. An Interface visibility must be public (or) none.

If we add a new method to an abstract class then we have the option of providing default implementation and therefore all the existing code might work properly If we add a new method to an Interface then we have to track down all the implementations of the interface and define implementation for the new method
An abstract class can contain constructors An Interface cannot contain constructors
Abstract classes are fast Interfaces are slow as it requires extra indirection to find the corresponding method in the actual class.

31. What are the important benefits of using Hibernate Framework?
Answer:
Some of the important benefits of using hibernate framework are:
Hibernate eliminates all the boiler-plate code that comes with JDBC and takes care of managing resources, so we can focus on business logic.
Hibernate framework provides support for XML as well as JPA annotations, that makes our code implementation independent.
Hibernate provides a powerful query language (HQL) that is similar to SQL. However, HQL is fully object-oriented and understands concepts like inheritance, polymorphism, and association.
Hibernate is an open source project from Red Hat Community and used worldwide. This makes it a better choice than others because the learning curve is small and there are tons of online documentation and help is easily available in forums.
Hibernate is easy to integrate with other Java EE frameworks, it’s so popular that Spring Framework provides built-in support for integrating hibernate with Spring applications.
Hibernate supports lazy initialization using proxy objects and perform actual database queries only when it’s required.
Hibernate cache helps us in getting better performance.
For database vendor-specific feature, hibernate is suitable because we can also execute native sql queries.
Overall hibernate is the best choice in the current market for ORM tool, it contains all the features that you will ever need in an ORM tool.

32. What are the differences between getting and load method?
Answer:
The differences between getting () and load() methods are given below.
No. get() load()
1) Returns null if the object is not found. Throws ObjectNotFoundException if the object is not found.
2) get() method always hit the database. load() method doesn’t hit the database.
3) It returns a real object, not a proxy. It returns a proxy object.
4) It should be used if you are not sure about the existence of instance. It should be used if you are sure that the instance exists.

33. What are the advantages of Hibernate over JDBC?
Answer:
Some of the important advantages of Hibernate framework over JDBC are:
Hibernate removes a lot of boiler-plate code that comes with JDBC API, the code looks cleaner

and readable.
Hibernate supports inheritance, associations, and collections. These features are not present with JDBC API.
Hibernate implicitly provides transaction management, in fact most of the queries can't be executed outside transaction. In JDBC API, we need to write code for transaction management using commit and rollback.
JDBC API throws SQLException that is a checked exception, so we need to write a lot of try-catch block code. Most of the times it's redundant in every JDBC call and used for transaction management. Hibernate wraps JDBC exceptions and throw JDBCException or HibernateException un-checked exception, so we don't need to write code to handle it. Hibernate built-in transaction management removes the usage of try-catch blocks.
Hibernate Query Language (HQL) is more object oriented and close to java programming language. For JDBC, we need to write native sql queries.
Hibernate supports caching that is better for performance, JDBC queries are not cached hence performance is low.
Hibernate provide option through which we can create database tables too, for JDBC tables must exist in the database.
Hibernate configuration helps us in using JDBC like connection as well as JNDI DataSource for the connection pool.

This is a very important feature in enterprise application and completely missing in JDBC API.
Hibernate supports JPA annotations, so code is independent of the implementation and easily replaceable with other ORM tools. JDBC code is very tightly coupled with the application.
In case you are facing any challenges with these java interview questions, please comment on your problems in the section below. Apart from this Java Interview Questions Blog, if you want to get trained from professionals on this technology, you can opt for structured training from edureka! Click below to know more.

34. What are the life-cycle methods for a jsp?
Answer:
Method Description
public void jspInit() It is invoked only once, same as init method of servlet.
public void _jspService(ServletRequest request,ServletResponse)throws ServletException,IOException It is invoked at each request, same as service() method of servlet.
public void jspDestroy() It is invoked only once, same as destroy() method of servlet.

35. What are the JSP implicit objects?
Answer:
JSP provides 9 implicit objects by default.

They are as follows:
Object Type
1) out JspWriter
2) request HttpServletRequest
3) response HttpServletResponse

4) config ServletConfig
5) session HttpSession
6) application ServletContext
7) pageContext PageContext
8) page Object
9) exception Throwable

36. What is the association?
Answer: Association is a relationship where all object have their own lifecycle and there is no owner. Let's take the example of Teacher and Student. Multiple students can associate with a single teacher and a single student can associate with multiple teachers but there is no ownership between the objects and both have their own lifecycle. This relationship can be one to one, One to many, many to one and many to many.

37. What do you mean by aggregation?
Answer: an Aggregation is a specialized form of Association where all object have their own lifecycle but there is ownership and child object can not belong to another parent object. Let's take an example of Department and teacher. A single teacher can not belong to multiple departments, but if we delete the department teacher object will not destroy.

38. What if the main() method is declared as private? What happens when the static modifier is removed from the signature of the main() method?
Answer: When the main() method is declared as private, the program compiles but during runtime it shows "main() method not public." message.
When the static modifier is removed from the signature of the main() method, the Program compiles but at runtime throws an error "NoSuchMethodError".

39.Can we rethrow the same exception from catch handler?
Answer: Yes we can rethrow the same exception from our catch handler. If we want to rethrow checked exception
from a catch block we need to declare that exception.

40. What is the first argument of the String array in main() method?
Answer: Unlike in C/C++ where the first element by default is the program name, the string array in main() has no element, the String array is empty.

41. What are the different ways of implementing thread? Which one is more advantageous?
Answer:
The thread can be implemented by
Using the runnable interface
Inheriting from the Thread class.
The use of the Runnable interface is more advantageous because when going for multiple inheritances, the only interface can help.

42. Why Java doesn’t support multiple inheritance ?
Answer: Java doesn’t support multiple inheritance in classes because of “Diamond Problem”. To know more about diamond problem with example, read Multiple Inheritance in Java. However multiple inheritance is supported in interfaces. An interface can extend multiple interfaces because they just declare the methods and implementation will be present in the implementing class. So there is no issue of diamond problem with interfaces.

43. What are the differences between include directive and include action?
Answer:
include directive include action
The include directive includes the content at page translation time. They include action includes the content at request time.
The include directive includes the original content of the page so page size increases at runtime.

They include action doesn’t include the original content rather invokes the include() method of Vendor provided class.
It’s better for static pages. It’s better for dynamic pages.

44. What are the different tags provided in JSTL?
Answer:
There are 5 types of JSTL tags.
core tags
sql tags
xml tags
internationalization tags
functions tags

45. How to delete a Cookie in a JSP ?
Answer:
The following code explain how to delete a Cookie in a JSP:
Cookie mycook = new Cookie(“name1″,”value1”);
response.addCookie(mycook1);
Cookie killmycook = new Cookie(“mycook1″,”value1”);
killmycook . set MaxAge (0);
killmycook . set Path (“/”);
killmycook . addCookie (killmycook 1);

46. Why should we not configure JSP standard tags in web.xml?
Answer: We don’t need to configure JSP standard tags in web.xml because when container loads the web application and find TLD files, it automatically configures them to be used directly in the application JSP pages. We just need to include it in the JSP page using taglib directive.
In case you are facing any challenges with these java interview questions, please comment on your problems in the section below. Apart from this Java Interview Questions Blog, if you want to get trained from professionals on this technology.

47. For searches. Which one is most preferred: Array List or Linked List?
Answer:
ArrayList. Searching an element is faster in ArrayList compared to LinkedList.

48. What is the difference between Iterator and Enumeration ?
Answer:
1) Iterator allows removing elements from the underlying collection during the iteration using its remove() method. We cannot add/remove elements from a collection when using enumerator.
2) Iterator has improved method names.
Enumeration.hasMoreElement() -> Iterator.hasNext()
Enumeration.nextElement() -> Iterator.next().

49. What is the difference between Iterator and List Iterator?
Answer:
Following are the major differences between them:
1) Iterator can be used for traversing Set, List, and Map. ListIterator can only be used for traversing a List.
2) We can traverse only in the forward direction using Iterator. ListIterator can be used for traversing in both the directions(forward and backward).

50. Explain the jspDestroy() method?
Answer: jspDestry() method is invoked from javax.servlet.jsp.JspPage interface whenever a JSP page is about to be destroyed. Servlets destroy methods can be easily overridden to perform cleanup, like when closing a database connection.

51. What is Hibernate Framework?
Answer: Object-relational mapping or ORM is the programming technique to map application domain model objects to the relational database tables. Hibernate is java based ORM tool that provides framework for mapping application domain objects to the relational database tables and vice versa.
Hibernate provides a reference implementation of the Java Persistence API, that makes it a great choice as an ORM tool with benefits of loose coupling. We can use the Hibernate persistence API for CRUD operations. Hibernate framework provides the option to map plain old java objects to traditional database tables with the use of JPA annotations as well as XML based configuration.
Similarly hibernate configurations are flexible and can be done from XML configuration file as well as programmatically.

52. What is the difference between sleep and wait in Java?
Answer: Both are used to pause thread that is currently running, however, sleep() is meant for short pause because it does not release the lock, while wait() is meant for the conditional wait. This is why it releases the lock, which can then be developed by a different thread to alter the condition of which it is waiting.

53. Describe what a thread-local variable is in Java?

Answer: Thread-local variables are variables restricted to a thread. It is like thread's own copy which is not shared between a multitude of threads. Java offers a ThreadLocal class to upkeep thread-local variables. This is one of the many ways to guarantee thread-safety.

However, it is important to be mindful while using a thread-local variable in a controlled environment, e.g. with web servers where worker thread outlives any application variable.

Any thread local variable which is not taken away once its work is done can hypothetically cause a memory leak in Java application.