JAVA INTERVIEW QUESTIONS

1. **What do you know about Java?**

Java is a high-level programming language originally developed by Sun Microsystems and released in 1995. Java runs on a variety of platforms, such as Windows, Mac OS, and the various versions of UNIX.

A **high-level programming language** is a programming language designed to simplify computer programming, having a strong abstraction from the details of the computer. Its source code contains easy-to-read syntax that is later converted into a low-level language, which can be recognized and run by a specific CPU.

2. **What are the supported platforms by Java Programming Language?**

Java runs on a variety of platforms, such as Windows, Mac OS, and the various versions of UNIX/Linux like HP-Unix, Sun Solaris, Redhat Linux, Ubuntu, CentOS, etc.

3. **List any five features of Java?**

- ***Object Oriented*** – everything is java in an object.
- ***Platform Independent*** – Java does not depend on any type of platform, it's a "write once, run anywhere" language. In Java, programs are compiled into byte code and that byte code is platform-independent.

- ***Compiled and Interpreted*** – java can be considered both a compiled and an interpreted language because its source code is first compiled into a binary byte-code. This byte-code runs on the Java Virtual Machine (JVM), which is usually a software-based interpreter.

- ***Secured*** – Java is best known for its security. With Java we can develop virus-free systems.

- ***Robust*** – it means strong. Java is robust because:
  - It uses strong memory management.
  - There is a lack of pointers that avoids security problems.
  - There is automatic garbage collection in java which runs on the Java Virtual Machine to get rid of objects which are not being used by a Java application anymore.
  - There are exception handling and the type checking mechanism in Java.

4. **Why is Java considered dynamic?**

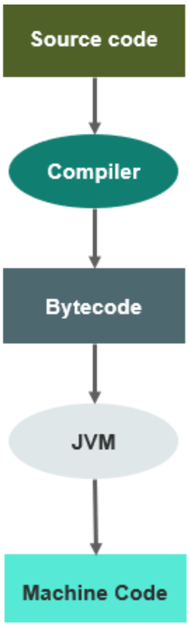
Because Java is designed to adapt to an evolving environment. Java programs can carry extensive amount of run-time information that can be used to verify and resolve accesses to objects on run-time.

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

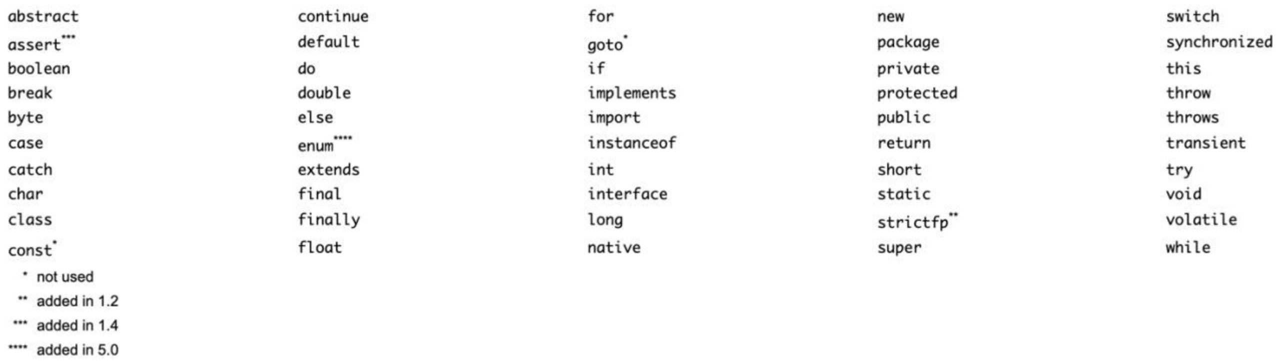
Java virtual machine (JVM) is a virtual machine that enables a computer to run Java programs as well as programs written in other languages that are also compiled to Java bytecode.

In Java, programs are compiled into byte code and that byte code is platform-independent. Bytecode in Java is the reason why java is platform-independent, as soon as a Java program is compiled bytecode is generated. This byte code is distributed over the web and interpreted by virtual Machine (JVM) on whichever platform it is being run.

Bytecode is program code that has been compiled from source code into low-level code designed for a software interpreter. It may be executed by a virtual machine (such as a JVM) or further compiled into machine code, which is recognized by the processor.



6. List some Java keywords.



abstract
assert***
boolean
break
byte
case
catch
char
class
const*

continue
default
do
double
else
enum****
extends
final
finally
float

for
goto*
if
implements
import
instanceof
int
interface
long
native

new
package
private
protected
public
return
short
static
strictfp**
super

switch
synchronized
this
throw
throws
transient
try
void
volatile
while

* not used
** added in 1.2
*** added in 1.4
**** added in 5.0

7. List Java Data types?

a) Primitive Types – for storing simple values (numbers, characters, booleans)

TYPE	BYTES	Example
byte	1	[-128, 127] ***range***
short	2	[-32K, 32K] ***range***
int	4	[-2B, 2B] ***range***
long	8	
float	4	
double	8	3.5; 4,6
char	2	a, b, c,…
boolean	1	true / false

b) Reference (Non-primitive) – for storing complex objects (array, object, date, string)

8. What is the default value of byte datatype in Java?
Default value of byte datatype is 0.

9. What is the default value of float and double datatype in Java?
For float is 0.0f and for double is 0.0d.

10. When a byte datatype is used?
This data type is used to save space in large arrays, mainly in place of integers, since a byte is four times smaller than an int.

11. What kind of variables a class can consist of?

a) ***Local variable*** – is a variable defined inside a function / method, constructor or block. The variable will be declared and initialized within the function and it will be destroyed when the function has completed.

b) ***Instance variable*** – is variables within a class but outside any method. These variables are instantiated when the class is loaded.

c) ***Class Variable*** – is a variable declared within a class, outside any method, with the ***static*** keyword. Class variables are also known as static variables.

12. What is Singleton class?
Singleton class control object creation, limiting the number to one but allowing the flexibility to create more objects if the situation changes.

13. List the three steps for creating an Object for a class?

1) object is first **declared**
2) then **instantiated**
3) and lastly **initialized**.

14. What do you mean by Access Modifier?
Access Modifier – is a special keyword that determines if other classes, variables, methods and constructor in this program can access these classes, variables, methods and constructor. Access modifiers serve to set the access levels.

Access modifiers (or access specifiers) is a special keyword in object-oriented languages that set the accessibility of classes, variables, methods, and other members.

Access modifier can be***:***

- **Public** for everything that should be exposed outside of the class.

- **Private** for hiding the implementation detail. Private methods are not inherited by subclasses, as they're not accessed outside of the class.

- **Protected** is like public but only within the current package, so as long as we are inside a package e.g. "***com.alainfonso***" this field/variable/cosntructor is treated like public. But they're also accessible by child classes in different packages. Protected is confusing and considered a bad practice and we should avoid it.

15. What do you mean by synchronized Non-Access-Modifier?
Java provides these modifiers for providing functionalities other than Access Modifiers, synchronized used to indicate that a method can be accessed by only one thread at a time.

16. Define Packages in Java?
A Package can be defined as a grouping of related types (classes, interfaces, enumerations and annotations) providing access protection and name space management. A ***package*** is a namespace (abstract container) that organizes a set of related classes and interfaces.

17. Why Packages are used?
Packages are used in Java in-order to prevent naming conflicts, to control access, to make searching/locating and usage of classes, interfaces, enumerations and annotations, etc., easier.

18. According to Java Operator precedence, which operator is considered to have with highest precedence (priority)?
Postfix operators i.e **() []** . is at the highest precedence.

19. Variables used in a switch statement can be used with which datatypes?
Variables used in a switch statement can only be a *string, enum, byte, short, int*, or *char*.

20. When parseInt() method can be used?
This method is used to get the primitive data type of a certain String.

21. Why is String class considered immutable?
The String class is immutable, so that once a String object is created, it cannot be changed. Since String is immutable, it can safely be shared between many threads, which is considered very important for multithreaded programming.

22. Why is StringBuffer called mutable?
If there is a necessity to make a lot of modifications to Strings of characters then StringBuffer should be used.

23. What is the difference between StringBuffer and StringBuilder class?
Use StringBuilder whenever possible because it is faster than StringBuffer. But, if thread safety is necessary then use StringBuffer objects.

24. What is finalize() method?
It is possible to define a method that will be called just before an object's final destruction by the garbage collector. This method is called finalize( ), and it can be used to ensure that an object terminates cleanly.

25. What is an Exception?
An exception is a problem that arises during the execution of a program. Exceptions are caught by handlers positioned along the thread's method invocation stack.

26. What are the three type of Exceptions?

- **Checked Exceptions –** is an exception that is typically a user error or a problem that cannot be foreseen by the programmer. For example, if a file is to be opened, but the file cannot be found, an exception occurs. These exceptions cannot simply be ignored at the time of compilation.

- **Runtime Exception** or **Unchecked exception –** is type of exception that doesn't get checked by the compiler at compile time because of programming errors. Here are some examples of unchecked exceptions: NullPointerExcpetion, ArithmeticException, illegalArgumentException, indexOutBoundsException, illegalStateException.

- **Errors** – it indicates an error external to our application, e.g. stack overflow error or out of memory error.

27. Which are the two subclasses under Exception class?

The Exception class has two main subclasses:

- IOException class
- RuntimeException class

28. When is the ArrayStoreException thrown?

When copying elements between different arrays, if the source or destination arguments are not arrays or their types are not compatible, an ArrayStoreException will be thrown.

29. When ArithmeticException is thrown?

The ArithmeticException is thrown when integer is divided by zero or taking the remainder of a number by zero. It is never thrown in floating-point operations.

30. When throws keyword is used?

If a method does not handle a checked exception, the method must declare it using the throws keyword. The throws keyword appears at the end of a method's signature.

31. When is the *throw* keyword used?

An exception can be thrown, either a newly instantiated one or an exception that you just caught, by using throw keyword.

32. How is the *finally* keyword used under Exception Handling?

The ***finally*** keyword is used to create a block of code that follows a try block. A finally block of code always executes, whether or not an exception has occurred.

33. What things should be kept in mind while creating your own exceptions in Java?

- All exceptions must be a child of Throwable.
- If you want to write a checked exception that is automatically enforced by the Handle or Declare Rule, you need to extend the Exception class.
- You want to write a runtime exception; you need to extend the RuntimeException class.

34. When is the *super* keyword used?

If the method overrides one of its superclass's methods, overridden method can be invoked through the use of the keyword super. It can be also used to refer to a hidden field.

35. Define Inheritance?

Inheritance – it's a mechanism that enables an object to take on the properties and methods of another object, making it easy to reuse code in different parts of an application. This mechanism consists of defining the common behaviors/functions in one class and then inherit in other classes, allowing us to eliminate redundant code.

36. What is Polymorphism?

Polymorphism – *poly* means "many" and *morph* "form", "polymorphism" means many forms. It's the ability of an object to take on many forms. Polymorphism consists of overriding in this child class a method that has been created in the Parent Class.

It's a core concept of an object-oriented paradigm that provides a way to perform a single action in different forms.

37. What is Encapsulation?

Encapsulation is mechanism of grouping related variables and functions that operate on them into a single unit. Encapsulation is also referred to as data hiding, we hide the implementation details from users, making security or privacy to data possible. By means of this mechanism we can reduce complexity and increase code reusability.

38. What is Abstraction?

Abstraction is a technique that consists of hiding unnecessary information (some properties and methods) in our classes, and showing only the essential, and also reducing the impact of changes in the code. Abstraction is important because it's a good practice to have in our classes a minimal number of functions or methods exposed to the outside.

This technique brings the following benefits:

- ***Simpler Interface*** – using and understanding an object with a few properties and methods is easier than an object with several properties and methods.

- ***Reduce the Impact of Change*** – we can change these hidden methods and properties, and it won't impact the rest of the applications code.

39. Is there any limitation of using Inheritance?

Yes, since inheritance inherits everything from the super class and interface, it may make the subclass too clustering and sometimes error-prone when dynamic overriding or dynamic overloading in some situation.

40. What is Abstract class?

Abstract class is the type of class that cannot be instantiated and is either partially implemented or not at all implemented. This class contains one or more abstract methods which are simply method declarations without a body.

41. When Abstract methods are used?

Abstract methods are used in the situation we want a class to contain a particular method but we want the actual implementation of that method to be determined by child classes, so we declare the method in the parent class as abstract.

42. What is the primary benefit of Encapsulation?
The main benefit of encapsulation is the ability to modify our implemented code without breaking the code of others who use our code. Encapsulation gives maintainability, flexibility and extensibility to our code.

43. What is an Interface?
Interface is a collection of abstract methods, it's similar to a Class but only include method declarations, no implementation (it declares the list of methods that a Class should provide). We use interface to build loosely-coupled, extensible, testable applications.

44. Give some features of Interface?

- Interface cannot be instantiated
- An interface does not contain any constructors.
- All of the methods in an interface are abstract.

45. What is a thread?
Thread is a single sequential flow of control within a program in the other words the path followed when executing a program. In Java, creating a thread is accomplished by implementing an the *Runnable* interface and extending the *Thread* class.

46. What do you mean by Multithreaded program?
A multithreaded program contains two or more parts that can run concurrently (simultaneously). Each part of such a program is called a thread, and each thread defines a separate path of execution.

47. Explain garbage collection in Java?
Garbage collection in Java is s process done through a component called Garbage Collector that consists of freeing the memory by cleaning those objects with no references and are unused for certain period of time.

48. Define immutable object?
An immutable object can't be changed once it has being created.

49. Explain the usage of this() with constructors?
The ***this*** keyword with ***constructors*** is used to call another constructor in the same class. Doing so is called an explicit constructor invocation.

https://docs.oracle.com/javase/tutorial/java/javaOO/thiskey.html#:~:text=Within%20an%20instance%20method%20or,a%20constructor%20by%20using%20this%20.

50. Difference between throw and throws?

- ***throw*** is used to trigger an exception whereas ***throws*** is used in declaration of exception.
- Without throws, checked exception cannot be handled whereas checked exception can be propagated with throws.

51. Explain the following line used under Java Program –
public static void main (String args[ ])

- ***public*** − it is the access specifier.
- ***static*** – it means that it is a class variable, it allows main() to be called without instantiating a particular instance of a class.
- ***void*** – function/method type which returns no value.
- ***main()*** − this method is called at the beginning of a Java program.
- ***String args[ ]*** − args parameter is an instance array of class String

52. Define JRE i.e. Java Runtime Environment?
Java Runtime Environment is an implementation of the Java Virtual Machine which executes Java programs. It provides the minimum requirements for executing a Java application;

53. What is the difference between object-oriented programming language and object-based programming language?
Object based programming languages follow all the features of OOPs except Inheritance. JavaScript is an example of object based programming languages.

54. What is the purpose of default constructor?
The java compiler creates a default constructor only if there is no constructor in the class.

55. Can a constructor be made final?
No, this is not possible.

56. What is static block?
It is used to initialize the static data member, It is executed before main method at the time of class loading.

57. Define Composition?
Composition is the concept of holding the reference of the other class within some other class.

58. What is function overloading?
Method Overloading is when a class has multiple functions by same name but different parameters.

59. What is function overriding?
Method Overriding is when a subclass provides a specific implementation of a method that is already provided by its parent class.

60. Difference between Overloading and Overriding?

- Method overloading increases the readability of the program.
- Method overriding provides the specific implementation of the method that is already provided by its super class.
- Parameters must be different in case of overloading.
- Parameter must be same in case of overriding.

61. What restrictions are placed on method overriding?
Overridden methods must have the same name, argument list, and return type. The overriding method may not limit the access of the method it overrides.

62. What is final class?
Final classes are created so the methods implemented by that class cannot be overridden nor inherited.

63. What is the difference between static and non-static variables?
A static variable is associated with the class as a whole rather than with specific instances of a class. Non-static variables take on unique values with each object instance.

64. Explain the use of sub-class in a Java program?
Sub class inherits all the public and protected methods and the implementation. It also inherits all the default modifier methods and their implementation.

65. What's the difference between constructors and other methods?
Constructors must have the same name as the class and cannot return a value. They are only called once while regular methods could be called many times.

66. Can you call one constructor from another if a class has multiple constructors?
Yes, use this() syntax.

67. What's the difference between the methods sleep() and wait()?
The code sleep(2000); puts thread aside for exactly two seconds. The code wait(2000), causes a wait of up to two second. A thread could stop waiting earlier if it receives the notify() or notifyAll() call. The method wait() is defined in the class Object and the method sleep() is defined in the class Thread.

68. What is a transient variable?
A transient variable is a variable that may not be serialized during Serialization and which is initialized by its default value during de-serialization,

69. What is synchronization?
Synchronization is the capability to control the access of multiple threads to shared resources. synchronized keyword in java provides locking which ensures mutual exclusive access of shared resource and prevent data race.

70. What is the Collections API?
The Collections API is a set of classes and interfaces that support operations on collections of objects.

71. Does garbage collection guarantee that a program will not run out of memory?
Garbage collection does not guarantee that a program will not run out of memory. It is possible for programs to use up memory resources faster than they are garbage collected. It is also possible for programs to create objects that are not subject to garbage collection.

72. What is the difference between a break statement and a continue statement?
A break statement results in the termination of the statement to which it applies (switch, for, do, or while). A continue statement is used to end the current loop iteration and return control to the loop statement.

73. If a variable is declared as private, where may the variable be accessed?
A private variable may only be accessed within the class in which it is declared.

74. Which class is the immediate superclass of the Container class?
Component class is the immediate super class.

75. What class of exceptions are generated by the Java run-time system?
The Java runtime system generates RuntimeException and Error exceptions.

76. Under what conditions is an object's finalize() method invoked by the garbage collector?
The garbage collector invokes an object's finalize() method when it detects that the object has become unreachable.

77. Can try statements be nested?
Yes they can.

78. What are ClassLoaders?
A class loader is an object that is responsible for loading classes. The class ClassLoader is an abstract class.

79. What is the difference between an Interface and an Abstract class?
An abstract class can have instance methods that implement a default behavior. An Interface can only declare constants and instance methods, but cannot implement default behavior and all methods are implicitly abstract. An interface has all public members and no implementation.

80. What will happen if static modifier is removed from the signature of the main method?
Program throws "NoSuchMethodError" error at runtime.

81. What is the difference between error and an exception?
An error is an irrecoverable condition occurring at runtime. Such as OutOfMemory error. Exceptions are conditions that occur because of bad input etc. e.g. FileNotFoundException will be thrown if the specified file does not exist.

82. Is it necessary that each try block must be followed by a catch block?
It is not necessary that each try block must be followed by a catch block. It should be followed by either a catch block or a finally block.

83. What is the Locale class?
The Locale class is used to tailor program output to the conventions of a particular geographic, political, or cultural region.

84. What are synchronized methods and synchronized statements?
Synchronized methods are methods that are used to control access to an object. A synchronized statement can only be executed after a thread has acquired the lock for the object or class referenced in the synchronized statement.

85. Can constructor be inherited?
No, constructor cannot be inherited.

86. What are the advantages of ArrayList over arrays?
ArrayList can grow dynamically and provides more powerful insertion and search mechanisms than arrays.

87. Why deletion in LinkedList is fast than ArrayList?
Deletion in linked list is fast because it involves only updating the next pointer in the node before the deleted node and updating the previous pointer in the node after the deleted node.

88. How do you decide when to use ArrayList and LinkedList?
If you need to frequently add and remove elements from the middle of the list and only access the list elements sequentially, then LinkedList should be used. If you need to support random access, without inserting or removing elements from any place other than the end, then ArrayList should be used.

89. What is a Values Collection View?
It is a collection returned by the values() method of the Map Interface, It contains all the objects present as values in the map.

90. What is dot operator?
The dot operator(.) is used to access the instance variables and methods of class objects.It is also used to access classes and sub-packages from a package.

91. Where and how can you use a private constructor?
Private constructor is used if you do not want other classes to instantiate the object and to prevent subclassing.T

92. What is type casting?
Type casting means treating a variable of one type as though it is another type.

93. Describe life cycle of thread?

A thread is a execution in a program. The life cycle of a thread include:

- Newborn state
- Runnable state
- Running state
- Blocked state
- Dead state

94. What is the difference between the >> and >>> operators?
The >> operator carries the sign bit when shifting right. The >>> zero-fills bits that have been shifted out.

95. Which method of the Component class is used to set the position and size of a component?
setBounds() method is used for this purpose.

96. What is the range of the short type?
The range of the short type is -(2^15) to 2^15 - 1.

97. Does Java allow Default Arguments?
No, Java does not allow Default Arguments.

98. Where import statement is used in a Java program?
Import statement is allowed at the beginning of the program file after package statement.

99. Explain suspend() method under Thread class?
It is used to pause or temporarily stop the execution of the thread.

100. Explain isAlive() method under Thread class?
It is used to find out whether a thread is still running or not.

101. What is currentThread()?
It is a public static method used to obtain a reference to the current thread.

102. Which class is used by server applications to obtain a port and listen for client requests?
java.net.ServerSocket class is used by server applications to obtain a port and listen for client requests

103. Why Generics are used in Java?
Generics provide compile-time type safety that allows programmers to catch invalid types at compile time. Java Generic methods and generic classes enable programmers to specify, with a single method declaration, a set of related methods or, with a single class declaration, a set of related types.

104. Can an Interface extend another Interface?
Yes an Interface can inherit another Interface, for that matter an Interface can extend more than one Interface.

105. Which object-oriented Concept is achieved by using overloading and overriding?
Polymorphism

106. What is Downcasting?
It is the casting from a general to a more specific type, i.e. casting down the hierarchy.

107. What environment variables do I need to set on my machine in order to be able to run Java programs?
CLASSPATH and PATH are the two variables.

108. Is there any need to import java.lang package?
No, there is no need to import this package. It is by default loaded internally by the JVM.

109. What is Nested top-level class?
If a class is declared within a class and specify the static modifier, the compiler treats the class just like any other top-level class. Nested top-level class is an Inner class.

110. What is Externalizable interface?
Externalizable is an interface which contains two methods readExternal and writeExternal. These methods give you a control over the serialization mechanism.

111. If System.exit (0); is written at the end of the try block, will the finally block still execute?
No in this case the finally block will not execute because when you say System.exit (0); the control immediately goes out of the program, and thus finally never executes.

112. What is daemon thread?
Daemon thread is a low priority thread, which runs intermittently in the back ground doing the garbage collection operation for the java runtime system.

113. Which method is used to create the daemon thread?
setDaemon method is used to create a daemon thread.

114. Which method must be implemented by all threads?
All tasks must implement the run() method

115. What is the GregorianCalendar class?
The GregorianCalendar provides support for traditional Western calendars

116. What is the SimpleTimeZone class?
The SimpleTimeZone class provides support for a Gregorian calendar .

117. What is the difference between the size and capacity of a Vector?
The size is the number of elements actually stored in the vector, while capacity is the maximum number of elements it can store at a given instance of time.

118. Can a vector contain heterogenous objects?
Yes a Vector can contain heterogenous objects. Because a Vector stores everything in terms of Object.

119. What is an enumeration?
An enumeration is an interface containing methods for accessing the underlying data structure from which the enumeration is obtained. It allows sequential access to all the elements stored in the collection.

120. What is difference between Path and Classpath?
Path and Classpath are operating system level environment variales. Path is defines where the system can find the executables(.exe) files and classpath is used to specify the location of .class files.

121. Can a class declared as private be accessed outside its package?
No, it's not possible to accessed outside it's package.

122. What is the restriction imposed on a static method or a static block of code?
A static method should not refer to instance variables without creating an instance and cannot use "this" operator to refer the instance.

123. What is an object's lock and which object's have locks?
An object's lock is a mechanism that is used by multiple threads to obtain synchronized access to the object. A thread may execute a synchronized method of an object only after it has acquired the object's lock.

124. What is order of precedence and associativity and how are they used?
Order of precedence determines the order in which operators are evaluated in expressions. Associatity determines whether an expression is evaluated left-to-right or right-to-left.

125. If a method is declared as protected, where may the method be accessed?
A protected method may only be accessed by classes or interfaces of the same package or by subclasses of the class in which it is declared.

126. What is the difference between inner class and nested class?
When a class is defined within a scope of another class, then it becomes inner class. If the access modifier of the inner class is static, then it becomes nested class.

127. What is constructor chaining and how is it achieved in Java?
A child object constructor always first needs to construct its parent. In Java it is done via an implicit call to the no-args constructor as the first statement.

128. Can a double value be cast to a byte?
Yes, a double value can be cast to a byte.

129. How does a try statement determine which catch clause should be used to handle an exception?
When an exception is thrown within the body of a try statement, the catch clauses of the try statement are examined in the order in which they appear. The first catch clause that is capable of handling the exception is executed. The remaining catch clauses are ignored.

130. What will be the default values of all the elements of an array defined as an instance variable?
If the array is an array of primitive types, then all the elements of the array will be initialized to the default value corresponding to that primitive type.

131. Which package is used for pattern matching with regular expressions?
java.util.regex package is used for this purpose.

132. java.util.regex consists of which classes?
java.util.regex consists of three classes − Pattern class, Matcher class and PatternSyntaxException class.

133. Why is Java Architectural Neutral?
Java is architectural neutral because there are no implementation dependent features, for example, the size of primitive types is fixed.

134. How Java enabled High Performance?
Java uses Just-In-Time compiler to enable high performance. Just-In-Time compiler is a program that turns Java bytecode, which is a program that contains instructions that must be interpreted into instructions that can be sent directly to the processor.

Java is faster than the other interpreted languages, because Java bytecode is "close" to native code. It is still a little bit slower than a compiled language (e.g., C++).

135. What is JAR file?
JAR files is Java Archive fles and it aggregates many files into one. It holds Java classes in a library. JAR files are built on ZIP file format and have .jar file extension.

136. What is a WAR file?
This is Web Archive File and used to store XML, java classes, and JavaServer pages. which is used to distribute a collection of JavaServer Pages, Java Servlets, Java classes, XML files, static Web pages etc.

137. Define JIT compiler?
It improves the runtime performance of computer programs based on bytecode.

138. What is NullPointerException?
A NullPointerException is thrown when calling the instance method of a null object, accessing or modifying the field of a null object etc.

139. What are the ways in which a thread can enter the waiting state?
A thread can enter the waiting state by invoking its sleep() method, by blocking on IO, by unsuccessfully attempting to acquire an object's lock, or by invoking an object's wait() method. It can also enter the waiting state by invoking its (deprecated) suspend() method.

140. How does multi-threading take place on a computer with a single CPU?
The operating system's task scheduler allocates execution time to multiple tasks. By quickly switching between executing tasks, it creates the impression that tasks execute sequentially.

141. What invokes a thread's run() method?
After a thread is started, via its start() method of the Thread class, the JVM invokes the thread's run() method when the thread is initially executed.

142. Does it matter in what order catch statements for FileNotFoundException and IOException are written?
Yes, it does. The FileNoFoundException is inherited from the IOException. Exception's subclasses have to be caught first.

143. What is the difference between yielding and sleeping?

When a task invokes its yield() method, it returns to the ready state. When a task invokes its sleep() method, it returns to the waiting state.

144. Why Vector class is used?
The Vector class provides the capability to implement a growable array of objects. Vector proves to be very useful if you don't know the size of the array in advance, or you just need one that can change sizes over the lifetime of a program.

145. How many bits are used to represent Unicode, ASCII, UTF-16, and UTF-8 characters?
Unicode requires 16 bits and ASCII require 7 bits. Although the ASCII character set uses only 7 bits, it is usually represented as 8 bits. UTF-8 represents characters using 8, 16, and 18 bit patterns. UTF-16 uses 16-bit and larger bit patterns.

146. What are Wrapper classes?
These are classes that allow primitive types to be accessed as objects. Example: Integer, Character, Double, Boolean etc.

147. What is the difference between a Window and a Frame?
The Frame class extends Window to define a main application window that can have a menu bar.

148. Which package has light weight components?
javax.Swing package. All components in Swing, except JApplet, JDialog, JFrame and JWindow are lightweight components.

149. What is the difference between the paint() and repaint() methods?
The paint() method supports painting via a Graphics object. The repaint() method is used to cause paint() to be invoked by the AWT painting thread.

150. What is the purpose of File class?
It is used to create objects that provide access to the files and directories of a local file system.

151. What is the difference between the Reader/Writer class hierarchy and the InputStream/OutputStream class hierarchy?
The Reader/Writer class hierarchy is character-oriented, and the InputStream/OutputStream class hierarchy is byte-oriented.

152. Which class should you use to obtain design information about an object?
The Class class is used to obtain information about an object's design and java.lang.Class class instance represent classes, interfaces in a running Java application.

153. How to add menu shortcut to menu item?
If there is a button instance called b1, you may add menu short cut by calling b1.setMnemonic('F'), so the user may be able to use Alt+F to click the button.

154. Can you write a Java class that could be used both as an applet as well as an application?
Yes, just add a main() method to the applet.

155. What is the difference between Swing and AWT components?
AWT components are heavy-weight, whereas Swing components are lightweight. Heavy weight components depend on the local windowing toolkit. For example, java.awt.Button is a heavy weight component, when it is running on the Java platform for Unix platform, it maps to a real Motif button.

156. What is an applet?
An applet is a Java program that runs in a Web browser. An applet can be a fully functional Java application because it has the entire Java API at its disposal.

157. The immediate super class of the Applet class?
Panel is the immediate superclass. A panel provides space in which an application can attach any other component, including other panels.

158. Which Java operator is right associative?
The = operator is right associative.

159. An applet extends which class?
An applet extends java.applet.Applet class.

160. Explain Set Interface?
It is a collection of elements which cannot contain duplicate elements. The Set interface contains only methods inherited from Collection and adds the restriction that duplicate elements are prohibited.

161. Explain TreeSet?
It is a Set implemented when we want elements in a sorted order.

162. What is Comparable Interface?
It is used to sort collections and arrays of objects using the collections.sort() and java.utils. The objects of the class implementing the Comparable interface can be ordered.

163. What is Serialization and deserialization?
Serialization is the process of writing the state of an object to a byte stream. Deserialization is the process of restoring these objects.

164. What are use cases?
It is part of the analysis of a program and describes a situation that a program might encounter and what behavior the program should exhibit in that circumstance.

165. What is the relationship between clipping and repainting under AWT?
When a window is repainted by the AWT painting thread, it sets the clipping regions to the area of the window that requires repainting.
166. What is the purpose of the System class?
The purpose of the System class is to provide access to system resources.

167. How can a dead thread be restarted?
A dead thread cannot be restarted.

168. Which arithmetic operations can result in the throwing of an ArithmeticException?
Integer / and % can result in the throwing of an ArithmeticException.

169. Variable of the boolean type is automatically initialized as?
The default value of the boolean type is false.

170. What is the default value of an object reference declared as an instance variable?
Null, unless it is defined explicitly.

171. Can a top level class be private or protected?
No, a top level class can not be private or protected. It can have either "public" or no modifier.

172. Why do we need wrapper classes?
We can pass them around as method parameters where a method expects an object. It also provides utility methods.

173. When a thread is created and started, what is its initial state?
A thread is in the ready state as initial state after it has been created and started.

174. What is runtime polymorphism or dynamic method dispatch?
Runtime polymorphism or dynamic method dispatch is a process in which a call to an overridden method is resolved at runtime rather than at compile-time. In this process, an overridden method is called through the reference variable of a superclass.

175. What is Dynamic Binding(late binding)?
Binding refers to the linking of a procedure call to the code to be executed in response to the call. Dynamic binding means that the code associated with a given procedure call is not known until the time of the call at run-time.

176. Which number is denoted by leading zero in java?
Octal Numbers are denoted by leading zero in java, example: 06

177. Which number is denoted by leading 0x or 0X in java?
Hexadecimal Numbers are denoted by leading 0x or 0X in java, example − 0XF

178. Break statement can be used as labels in Java?
Yes, an example can be *break one;*

179. What is the immediate superclass of Menu?
MenuItem class

180. Explain main thread under Thread class execution?
The main thread is created automatically and it begins to execute immediately when a program starts. It ia thread from which all other child threads originate.

181. Life cycle of an applet includes which steps?

Life cycle involves the following steps:

- Initialization
- Starting
- Stopping
- Destroying
- Painting

182. Why is the role of init() method under applets?
It initializes the applet and is the first method to be called.

183. Which method is called by Applet class to load an image?
getImage(URL object, filename) is used for this purpose.

184. Define code as an attribute of Applet?
It is used to specify the name of the applet class.

185. Define canvas?
It is a simple drawing surface which are used for painting images or to perform other graphical operations.

186. Define Network Programming?
It refers to writing programs that execute across multiple devices (computers), in which the devices are all connected to each other using a network.

187. What is a Socket?
Sockets provide the communication mechanism between two computers using TCP. A client program creates a socket on its end of the communication and attempts to connect that socket to a server.

188. Advantages of Java Sockets?
Sockets are flexible and sufficient. Efficient socket based programming can be easily implemented for general communications. It cause low network traffic.

189. Disadvantages of Java Sockets?
Socket based communications allows only to send packets of raw data between applications. Both the client-side and server-side have to provide mechanisms to make the data useful in any way.

190. Which class represents the socket that both the client and server use to communicate with each other?
java.net.Socket class represents the socket that both the client and server use to communicate with each other.