1. What is a platform?

   A platform is the hardware or software environment in which a program runs. Most platforms can be described as a combination of the operating system and hardware, like Windows 2000/XP, Linux, Solaris, and MacOS.

---

2. What is the main difference between Java platform and other platforms?

   The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other hardware-based platforms.

   The Java platform has two components:

   1. The Java Virtual Machine (Java VM)
   2. The Java Application Programming Interface (Java API)

---

3. What is the Java Virtual Machine?

   The Java Virtual Machine is a software that can be ported onto various hardware-based platforms.

---

4. What is the Java API?

   The Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets.

---

5. What is the package?

   The package is a Java namespace or part of Java libraries. The Java API is grouped into libraries of related classes and interfaces; these libraries are known as packages.

---

6. What is native code?

   The native code is code that after you compile it, the compiled code runs on a specific hardware platform.

7. Is Java code slower than native code?

   Not really. As a platform-independent environment, the Java platform can be a bit slower than native code. However, smart compilers, well-tuned interpreters, and just-in-time bytecode compilers can bring performance close to that of native code without threatening portability.

---

8. What is the serialization?

   The serialization is a kind of mechanism that makes a class or a bean persistence by having its properties or fields and state information saved and restored to and from storage.

---

9. How to make a class or a bean serializable?

   By implementing either the java.io.Serializable interface, or the java.io.Externalizable interface. As long as one class in a class's inheritance hierarchy implements Serializable or Externalizable, that class is serializable.

---

10. How many methods in the Serializable interface?

    There is no method in the Serializable interface. The Serializable interface acts as a marker, telling the object serialization tools that your class is serializable.

---

11. How many methods in the Externalizable interface?

    There are two methods in the Externalizable interface. You have to implement these two methods in order to make your class externalizable. These two methods are readExternal() and writeExternal().

---

12. What is the difference between Serializalble and Externalizable interface?

    When you use Serializable interface, your class is serialized automatically by default. But you can override writeObject() and readObject() two methods to

control more complex object serailization process. When you use Externalizable interface, you have a complete control over your class's serialization process.

---

13. What is a transient variable?

A transient variable is a variable that may not be serialized. If you don't want some field to be serialized, you can mark that field transient or static.

---

14. Which containers use a border layout as their default layout?

The Window, Frame and Dialog classes use a border layout as their default layout.

---

15. How are Observer and Observable used?

Objects that subclass the Observable class maintain a list of observers. When an Observable object is updated it invokes the update() method of each of its observers to notify the observers that it has changed state. The Observer interface is implemented by objects that observe Observable objects.

---

16. What is synchronization and why is it important?

With respect to multithreading, synchronization is the capability to control the access of multiple threads to shared resources. Without synchronization, it is possible for one thread to modify a shared object while another thread is in the process of using or updating that object's value. This often causes dirty data and leads to significant errors.

---

17. What are synchronized methods and synchronized statements?

Synchronized methods are methods that are used to control access to an object. A thread only executes a synchronized method after it has acquired the lock for the method's object or class. Synchronized statements are similar to synchronized methods. A synchronized statement can only be executed after a thread has acquired the lock for the object or class referenced in the synchronized statement.

18. What are three 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 I/O, 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.

---

19. Can a lock be acquired on a class?

Yes, a lock can be acquired on a class. This lock is acquired on the class's Class object.

---

20. What's new with the stop(), suspend() and resume() methods in JDK 1.2?

The stop(), suspend() and resume() methods have been deprecated in JDK 1.2.

---

21. What is the preferred size of a component?

The preferred size of a component is the minimum component size that will allow the component to display normally.

---

22. What method is used to specify a container's layout?

The setLayout() method is used to specify a container's layout.

---

23. Which containers use a FlowLayout as their default layout?

The Panel and Applet classes use the FlowLayout as their default layout.

---

24. What is thread?

A thread is an independent path of execution in a system.

25. What is multithreading?

Multithreading means various threads that run in a system.

---

26. How does multithreading 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.

---

27. How to create multithread in a program?

You have two ways to do so. First, making your class "extends" *Thread* class. Second, making your class "implements" *Runnable* interface. Put jobs in a run() method and call start() method to start the thread.

---

28. Can Java object be locked down for exclusive use by a given thread?

Yes. You can lock an object by putting it in a "synchronized" block. The locked object is inaccessible to any thread other than the one that explicitly claimed it.

---

29. Can each Java object keep track of all the threads that want to exclusively access to it?

Yes.

---

30. What state does a thread enter when it terminates its processing?

When a thread terminates its processing, it enters the dead state.

---

31. 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.

---

32. What is the purpose of the wait(), notify(), and notifyAll() methods?

The wait(),notify(), and notifyAll() methods are used to provide an efficient way for threads to communicate each other.

---

33. What are the high-level thread states?

The high-level thread states are ready, running, waiting, and dead.

---

34. What is the Collections API?

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

---

35. What is the List interface?

The List interface provides support for ordered collections of objects.

---

36. How does Java handle integer overflows and underflows?

It uses those low order bytes of the result that can fit into the size of the type allowed by the operation.

---

37. What is the Vector class?

The Vector class provides the capability to implement a growable array of objects

---

38. What modifiers may be used with an inner class that is a member of an outer class?

    A (non-local) inner class may be declared as public, protected, private, static, final, or abstract.

---

39. 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.

---

40. What is an Iterator interface?

    The Iterator interface is used to step through the elements of a Collection.

---

41. 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.

---

42. 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.

---

43. Is sizeof a keyword?

    The sizeof operator is not a keyword in Java.

---

44. What are wrapped classes?

Wrapped classes are classes that allow primitive types to be accessed as objects.

---

45. Does garbage collection guarantee that a program will not run out of memory?

No, it doesn't. 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

---

46. What is the difference between preemptive scheduling and time slicing?

Under preemptive scheduling, the highest priority task executes until it enters the waiting or dead states or a higher priority task comes into existence. Under time slicing, a task executes for a predefined slice of time and then reenters the pool of ready tasks. The scheduler then determines which task should execute next, based on priority and other factors.

---

47. Name Component subclasses that support painting.

The Canvas, Frame, Panel, and Applet classes support painting.

---

48. What is a native method?

A native method is a method that is implemented in a language other than Java.

---

49. How can you write a loop indefinitely?

for(;;)--for loop; while(true)--always true, etc.

---

50. Can an anonymous class be declared as implementing an interface and extending a class?

An anonymous class may implement an interface or extend a superclass, but may not be declared to do both.

51. What is the purpose of finalization?

The purpose of finalization is to give an unreachable object the opportunity to perform any cleanup processing before the object is garbage collected.

---

52. Which class is the superclass for every class.

Object

---

53. What is the difference between the Boolean & operator and the && operator?

If an expression involving the Boolean & operator is evaluated, both operands are evaluated. Then the & operator is applied to the operand. When an expression involving the && operator is evaluated, the first operand is evaluated. If the first operand returns a value of true then the second operand is evaluated. The && operator is then applied to the first and second operands. If the first operand evaluates to false, the evaluation of the second operand is skipped.
Operator & has no chance to skip both sides evaluation and && operator does. If asked why, give details as above.

---

54. What is the GregorianCalendar class?

The GregorianCalendar provides support for traditional Western calendars.

---

55. What is the SimpleTimeZone class?

The SimpleTimeZone class provides support for a Gregorian calendar.

---

56. Which Container method is used to cause a container to be laid out and redisplayed?

validate()

57. What is the Properties class?

The properties class is a subclass of Hashtable that can be read from or written to a stream. It also provides the capability to specify a set of default values to be used.

---

58. What is the purpose of the Runtime class?

The purpose of the Runtime class is to provide access to the Java runtime system.

---

59. What is the purpose of the System class?

The purpose of the System class is to provide access to system resources.

---

60. What is the purpose of the finally clause of a try-catch-finally statement?

The finally clause is used to provide the capability to execute code no matter whether or not an exception is thrown or caught.

---

61. 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.

---

62. What must a class do to implement an interface?

It must provide all of the methods in the interface and identify the interface in its implements clause.

---

63. What is an abstract method?

An abstract method is a method whose implementation is deferred to a subclass. Or, a method that has no implementation (an interface of a method).

---

64. What is a static method?

A static method is a method that belongs to the class rather than any object of the class and doesn't apply to an object or even require that any objects of the class have been instantiated.

---

65. What is a protected method?

A protected method is a method that can be accessed by any method in its package and inherited by any subclass of its class.

---

66. What is the difference between a static and a non-static inner class?

A non-static inner class may have object instances that are associated with instances of the class's outer class. A static inner class does not have any object instances.

---

67. 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. All objects and classes have locks. A class's lock is acquired on the class's Class object.

---

68. When can an object reference be cast to an interface reference?

An object reference be cast to an interface reference when the object implements the referenced interface.

---

69. 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.

---

70. What do heavy weight components mean?

Heavy weight components like Abstract Window Toolkit (AWT), 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. In this relationship, the Motif button is called the peer to the java.awt.Button. If you create two Buttons, two peers and hence two Motif Buttons are also created. The Java platform communicates with the Motif Buttons using the Java Native Interface. For each and every component added to the application, there is an additional overhead tied to the local windowing system, which is why these components are called heavy weight.

---

71. Which package has light weight components?

javax.Swing package. All components in Swing, except JApplet, JDialog, JFrame and JWindow are lightweight components.

---

72. What are peerless components?

The peerless components are called light weight components.

---

73. What is the difference between the Font and FontMetrics classes?

The FontMetrics class is used to define implementation-specific properties, such as ascent and descent, of a Font object.

---

74. What happens when a thread cannot acquire a lock on an object?

If a thread attempts to execute a synchronized method or synchronized statement and is unable to acquire an object's lock, it enters the waiting state until the lock becomes available.

---

75. 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.

---

76. What classes of exceptions may be caught by a catch clause?

    A catch clause can catch any exception that may be assigned to the Throwable type. This includes the Error and Exception types.

---

77. What is the difference between throw and throws keywords?

    The *throw* keyword denotes a statement that causes an exception to be initiated. It takes the Exception object to be thrown as argument. The exception will be caught by an immediately encompassing try-catch construction or propagated further up the calling hierarchy.

    The *throws* keyword is a modifier of a method that designates that exceptions may come out of the mehtod, either by virtue of the method throwing the exception itself or because it fails to catch such exceptions that a method it calls may throw.

---

78. If a class is declared without any access modifiers, where may the class be accessed?

    A class that is declared without any access modifiers is said to have package or friendly access. This means that the class can only be accessed by other classes and interfaces that are defined within the same package.

---

79. What is the Map interface?

    The Map interface replaces the JDK 1.1 Dictionary class and is used associate keys with values.

---

80. Does a class inherit the constructors of its superclass?

A class does not inherit constructors from any of its superclasses.

---

81. Name primitive Java types.

The primitive types are byte, char, short, int, long, float, double, and boolean.

---

82. 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.

---

83. How can a GUI component handle its own events?

A component can handle its own events by implementing the required event-listener interface and adding itself as its own event listener.

---

84. How are the elements of a GridBagLayout organized?

The elements of a GridBagLayout are organized according to a grid. However, the elements are of different sizes and may occupy more than one row or column of the grid. In addition, the rows and columns may have different sizes.

---

85. What advantage do Java's layout managers provide over traditional windowing systems?

Java uses layout managers to lay out components in a consistent manner across all windowing platforms. Since Java's layout managers aren't tied to absolute sizing and positioning, they are able to accommodate platform-specific differences among windowing systems.

---

86. What are the problems faced by Java programmers who don't use layout managers?

Without layout managers, Java programmers are faced with determining how their GUI will be displayed across multiple windowing systems and finding a

common sizing and positioning that will work within the constraints imposed by each windowing system.

---

87. 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.

---

88. 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.

---

89. What is the purpose of the File class?

The File class is used to create objects that provide access to the files and directories of a local file system.

---

90. What restrictions are placed on method overloading?

Two methods may not have the same name and argument list but different return types.

---

91. 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. The overriding method may not throw any exceptions that may not be thrown by the overridden method.

---

92. What is casting?

There are two types of casting, casting between primitive numeric types and casting between object references. Casting between numeric types is used to convert larger values, such as double values, to smaller values, such as byte values. Casting between object references is used to refer to an object by a compatible class, interface, or array type reference.

---

93. Name Container classes.

Window, Frame, Dialog, FileDialog, Panel, Applet, or ScrollPane

---

94. What class allows you to read objects directly from a stream?

The ObjectInputStream class supports the reading of objects from input streams.

---

95. How are this() and super() used with constructors?

this() is used to invoke a constructor of the same class. super() is used to invoke a superclass constructor.

---

96. How is it possible for two String objects with identical values not to be equal under the == operator?

The == operator compares two objects to determine if they are the same object in memory. It is possible for two String objects to have the same value, but located indifferent areas of memory.

---

97. What an I/O filter?

An I/O filter is an object that reads from one stream and writes to another, usually altering the data in some way as it is passed from one stream to another.

---

98. What is the Set interface?

The Set interface provides methods for accessing the elements of a finite mathematical set. Sets do not allow duplicate elements.

---

99. What is the List interface?

The List interface provides support for ordered collections of objects.

---

100. What is the purpose of the enableEvents() method?

The enableEvents() method is used to enable an event for a particular object. Normally, an event is enabled when a listener is added to an object for a particular event. The enableEvents() method is used by objects that handle events by overriding their event-dispatch methods.

---

101. What is the difference between the File and RandomAccessFile classes?

The File class encapsulates the files and directories of the local file system. The RandomAccessFile class provides the methods needed to directly access data contained in any part of a file.

---

102. What interface must an object implement before it can be written to a stream as an object?

An object must implement the Serializable or Externalizable interface before it can be written to a stream as an object.

---

103. What is the ResourceBundle class?

The ResourceBundle class is used to store locale-specific resources that can be loaded by a program to tailor the program's appearance to the particular locale in which it is being run.

---

104. What is the difference between a Scrollbar and a ScrollPane?

A Scrollbar is a Component, but not a Container. A ScrollPane is a Container. A ScrollPane handles its own events and performs its own scrolling.

---

105. What is a Java package and how is it used?

A Java package is a naming context for classes and interfaces. A package is used to create a separate name space for groups of classes and interfaces. Packages are also used to organize related classes and interfaces into a single API unit and to control accessibility to these classes and interfaces.

---

106. What are the Object and Class classes used for?

The Object class is the highest-level class in the Java class hierarchy. The Class class is used to represent the classes and interfaces that are loaded by a Java program.

---

107. 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.

---

108. what is tunnelling?

Tunnelling is a route to somewhere. For example, RMI tunnelling is a way to make RMI application get through firewall. In CS world, tunnelling means a way to transfer data.

---

109. Does the code in finally block get executed if there is an exception and a return statement in a catch block?

If an exception occurs and there is a return statement in catch block, the finally block is still executed. The finally block will not be executed when the System.exit(1) statement is executed earlier or the system shut down earlier or the memory is used up earlier before the thread goes to finally block.

---

110. How you restrict a user to cut and paste from the html page?

Using javaScript to lock keyboard keys. It is one of solutions.

---

111. Is Java a super set of JavaScript?

No. They are completely different. Some syntax may be similar.

---

112. What is a Container in a GUI?

A Container contains and arranges other components (including other containers) through the use of layout managers, which use specific layout policies to determine where components should go as a function of the size of the container.

---

113. How the object oriented approach helps us keep complexity of software development under control?

We can discuss such issue from the following aspects:

- Objects allow procedures to be encapsulated with their data to reduce potential interference.
- Inheritance allows well-tested procedures to be reused and enables changes to make once and have effect in all relevant places.
- The well-defined separations of interface and implementation allows constraints to be imposed on inheriting classes while still allowing the flexibility of overriding and overloading.

---

114. What is polymorphism?

Polymorphism allows methods to be written that needn't be concerned about the specifics of the objects they will be applied to. That is, the method can be specified at a higher level of abstraction and can be counted on to work even on objects of yet unconceived classes.

---

115. What is design by contract?

The design by contract specifies the obligations of a method to any other methods that may use its services and also theirs to it. For example, the preconditions specify what the method required to be true when the method is called. Hence making sure that preconditions are. Similarly, postconditions specify what must be true when the method is finished, thus the called method has the responsibility of satisfying the post conditions.

In Java, the exception handling facilities support the use of design by contract, especially in the case of checked exceptions. The assert keyword can be used to make such contracts.

---

116. What are use cases?

A use case describes a situation that a program might encounter and what behavior the program should exhibit in that circumstance. It is part of the analysis of a program. The collection of use cases should, ideally, anticipate all the standard circumstances and many of the extraordinary circumstances possible so that the program will be robust.

---

117. What is the difference between interface and abstract class?

- interface contains methods that must be abstract; abstract class may contain concrete methods.
- interface contains variables that must be static and final; abstract class may contain non-final and final variables.
- members in an interface are public by default, abstract class may contain non-public members.
- interface is used to "implements"; whereas abstract class is used to "extends".
- interface can be used to achieve multiple inheritance; abstract class can be used as a single inheritance.
- interface can "extends" another interface, abstract class can "extends" another class and "implements" multiple interfaces.
- interface is absolutely abstract; abstract class can be invoked if a main() exists.
- interface is more flexible than abstract class because one class can only "extends" one super class, but "implements" multiple interfaces.
- If given a choice, use interface instead of abstract class.
