QUESTIONS
Oracle

Q1. What is PL/SQL ?

A1. A programming language available in Oracle. It is a way of wrapping SQL with 3GL constructs such as IF..THEN..ELSE, WHILE etc.

Q2. What is locking and what is it used for ?

A2 Locking is a mechanism to restrict access to data. It prevents one user updating data whilst another user is looking at it.

Q3. What types of locking are available with Oracle?

A3. Row locking and table locking.

Q4. What happens if a tablespace clause is left off of a primary key constraint clause?

A4. This results in the index that is automatically generated being placed in the users default tablespace. Since this will usually be the same tablespace as the table is being created in, this can cause serious performance problems.

Q5. What packages (if any) has Oracle provided for use by developers?

A5. Oracle provides the DBMS_ series of packages. There are many which developers should be aware of such as DBMS_SQL, DBMS_PIPE, DBMS_TRANSACTION, DBMS_LOCK, DBMS_ALERT, DBMS_OUTPUT, DBMS_JOB, DBMS_UTILITY, DBMS_DDL, UTL_FILE.

Q6. When should more than one DB writer process be used? How many should be used?

A6. If the UNIX system being used is capable of asynchronous IO then only one is required, if the system is not capable of asynchronous IO then up to twice the number of disks used by Oracle number of DB writers should be specified by use of the db_writers initialisation parameter.

SQL

Q1. Explain the use of the WHERE clause.

A1. It directs SQL to extract data from rows where the value of the column is the same as the current value of the WHERE clause variable.

Q2. What technique is used to retrieve data from more than one table in a single SQL statement?

A2. The Join statement combines data from more that two tables

Q3. What is a foreign key?

A3. It identifies a related row in another table and establishes a logical relationship between rows in two tables.

Q4. What are the reasons for adding an index to a table ?

A4. To increase lookup performance. For a unique key to guarantee the uniqueness of the values for a column on a table.

Q5. What is a primary key on a table used for ?

A5. Guaranteeing the uniqueness of a column on a table.

Q6. What is the purpose of a view?

A6. Views can be used to join and display data from multiple tables and to limit the number of columns and/or rows that are selected and displayed.

C / Programming Languages

Q1. What is the difference between a local and a global variable ?

C / Programming Languages

A1. A local variable is visible only to the routine it is declared in. It is created on entry to the sub-routine and is destroyed on exit from the routine. A global variable is visible to all the routines in a program.

Q2. Describe the difference between a variable passed to a sub-routine by reference and a variable passed to a sub-routine by value.

A2. A variable passed by value cannot be modified by the sub-routine and the new value reflected back in the calling code.

Q3. What are the advantages of laying out code to a company standard ?

A3. There is no fixed answer to this but readability and maintainability should get mentioned somewhere.

Q4. What is the name of the entry function to every C program ?

A4. main

Q5. What are the arguments commonly passed to main ? What do they hold ?

A5. argc is the number of command line argument, argv is a pointer to an array holding the command line arguments.

Q6. What routines are used to program signal handling into a C program.

A6 signal() and sigset().

Q7. Name three types of inter-process communication mechanisms available to C programmers ?

A7. Pipes, Semaphores , Shared Memory, Message Queues.

UNIX

Q1. What is the most common editor used in the UNIX environment ?

A1. vi

Q2. In which file are the UNIX user details and passwords kept ?

A2. /etc/passwd

Q3. What is the purpose of the /tmp directory ?

A3. It is used for temporary files produced as a result of running commands and piping data to other commands.

Q4. What command is used to change the permissions on files and directories ?

A4. chmod

Q5. List three types of shells commonly available in a UNIX environment

A5. csh (C-Shell), ksh (Korn-Shell), sh (Bourne Shell), bash (Born Again Shell)

Java

Q1. All Object Oriented Languages share three common defining traits, explain these.

A1. Encapsulation, Polymorphism and Inheritance.

Q2. What is the difference between a class and a structure?

A2. In a class, members are private by default. In a structure, members are public by default.

Q3. What is the difference between a class and an object.

A3. A class defines an Objects methods and instance variables, i.e. the data it will store and methods to access that data. An Object is an instantiated Class which has a state.

Q4. What are the differences between an applet and an application.

A4. An application has no security restrictions, and is run like a normal application using java. An applet has security restrictions and is run either in an applet viewer or Web Browser. There is no main, as the applet viewer/Web Browser is the application and it instantiates the applet class instance you provide it.

Q5. What is swing.

A5. Swing is an enhanced version of AWT that provides light weight graphical components. These components are suitable for Applets that are going to be run within Web Browsers.

Q6. What are EJB's and how can they be used.

A6 Enterprise Java Beans come in two flavours, Entity and Session. The Entity EJB is an Object that has data like a record in a database. A Session EJB comes in two flavours, Stateful or Stateless, and this type of EJB is used for implementing Business rules.

CORE JAVA QUESTIONS by Systems Technology Group

1.what is a transient variable?
A transient variable is a variable that may not be serialized.

2.which containers use a border Layout as their default layout?
The window, Frame and Dialog classes use a border layout as their default layout.

3.Why do threads block on I/O?
Threads block on i/o (that is enters the waiting state) so that other threads may execute while the i/o Operation is performed.

4. 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.

5. 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 leads to significant errors.

6. 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.

7. 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.

8. Is null a keyword?
The null value is not a keyword.

9. 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.

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

11. Which containers use a FlowLayout as their default layout?
The Panel and Applet classes use the FlowLayout as their default layout.

12. What state does a thread enter when it terminates its processing?
When a thread terminates its processing, it enters the dead state.

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

14. Which characters may be used as the second character of an identifier,
but not as the first character of an identifier?
The digits 0 through 9 may not be used as the first character of an identifier but they may be used after the first character of an identifier.

15. What is the List interface?
The List interface provides support for ordered collections of objects.

16. 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.

17. What is the Vector class?
The Vector class provides the capability to implement a growable array of objects

18. 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.

19. What is an Iterator interface?
The Iterator interface is used to step through the elements of a Collection.

20. 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.

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

22. 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.

23What 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.

24. Which java.util classes and interfaces support event handling?
The EventObject class and the EventListener interface support event processing.

25. Is sizeof a keyword?
The sizeof operator is not a keyword.

26. What are wrapped classes?
Wrapped classes are classes that allow primitive types to be accessed as objects.

27. 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

28. What restrictions are placed on the location of a package statement
within a source code file?
A package statement must appear as the first line in a source code file (excluding blank lines and comments).

29. Can an object's finalize() method be invoked while it is reachable?
An object's finalize() method cannot be invoked by the garbage collector while the objectis still reachable. However, an object's finalize() method may be invoked by other objects.

30. What is the immediate superclass of the Applet class?
Panel

31. 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.

32. Name three Component subclasses that support painting.
The Canvas, Frame, Panel, and Applet classes support painting.

33. What value does readLine() return when it has reached the end of a file?
The readLine() method returns null when it has reached the end of a file.

34. What is the immediate superclass of the Dialog class?
Window

35. What is clipping?
Clipping is the process of confining paint operations to a limited area or shape.

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

37. Can a for statement loop indefinitely?
Yes, a for statement can loop indefinitely. For example, consider the following:
for(;;) ;

38. What are 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

39. When a thread blocks on I/O, what state does it enter?
A thread enters the waiting state when it blocks on I/O.

40. To what value is a variable of the String type automatically initialized?
The default value of an String type is null.

41. What is the catch or declare rule for method declarations?
If a checked exception may be thrown within the body of a method, the method must either catch the exception or declare it in its throws clause.

42. What is the difference between a MenuItem and a CheckboxMenuItem?
The CheckboxMenuItem class extends the MenuItem class to support a menu item that may be checked or unchecked.

43. What is a task' s priority and how is it used in scheduling?
A task' s priority is an integer value thatdentifies the relative order in which it should be executed with respect to other tasks. The scheduler attempts to schedule higher priority tasks before lower priority tasks.

44. What class is the top of the AWT event hierarchy?
The java.awt.AWTEvent class is the highest-level class in the AWT event-class hierarchy.

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

46. 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.

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

48. What is the range of the char type?
The range of the char type is 0 to 2^16 - 1.

49. In which package are most of the AWT events that support the event-delegation model defined?
Most of the AWT-related events of the event-delegation model are defined in the java.awt.event package. The AWTEvent class is defined in the java.awt package.

50. What is the immediate superclass of Menu?
MenuItem

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 immediate superclass of the MenuComponent class.
Object

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

54. 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.

55. Name three subclasses of the Component class.
Box.Filler, Button, Canvas, Checkbox, Choice, Container, Label, List, Scrollbar, or TextComponent

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

57. Which Container method is used to cause a container to be laid out and redisplayed?
validate()

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. How many times may an object's finalize() method be invoked by the garbage collector?
An object's finalize() methodmay only be invoked once by the garbage collector.

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 argument type of a program's main() method?
A program's main() method takes an argument of the String[] type.

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

63. 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.

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

65. 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.

66. 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.

67. What method is invoked to cause an object to begin executing as a separate thread?
The start() method of the Thread class is invoked to cause an object to begin executing as a separate thread.

68. Name two subclasses of the TextComponent class.
TextField and TextArea

69. What is the advantage of the event-delegation model over the earlier event-inheritance model?
The event-delegation model has two advantages over the event-inheritance model. First, it enables event handling to be handled by objects other than the ones that generate the events (or their containers). This allows a clean separation between a component's design and its use. The other advantage of the eventdelegation model is that it performs much better in applications where many events are generated. This performance improvement is due to the fact that the event-delegation model does not have to repeatedly process unhandled events, as is the case of the event-inheritance model.

70. Which containers may have a MenuBar?
Frame

71. How are commas used in the intialization and iteration parts of a for statement?
Commas are used to separate multiple statements within the initialization and iteration parts of a for statement.

72. 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 wait for a shared resource. When a thread executes an object's wait() method, it enters the waiting state. It only enters the ready state after another thread invokes the object's notify() or notifyAll() methods.

73. What is an abstract method?
An abstract method is a method whose implementation is deferred to a subclass.

74. How are Java source code files named?
A Java source code file takes the name of a public class or interface that is defined within the file. A source code file may contain at most one public class or interface. If a public class or interface is defined within a source code file, then the source code file must take the name of the public class or interface. If no public class or interface is defined within a source code file, then the file must take on a name that is different than its classes and interfaces. Source code files use the .java extension.

75. What is the relationship between the Canvas class and the Graphics class?
A Canvas object provides access to a Graphics object via its paint() method.

76. What are the high-level thread states?
The high-level thread states are ready, running, waiting, and dead.

77. What value does read() return when it has reached the end of a file?
The read() method returns -1 when it has reached the end of a file.

78. Can a Byte object be cast to a double value?
No, an object cannot be cast to a primitive value.

79. 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.

80. What is the difference between the String and StringBuffer classes?
String objects are constants. StringBuffer objects are not.

81. 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.

82. 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.

83. What is the Dictionary class?
The Dictionary class provides the capability to store key-value pairs.

84. How are the elements of a BorderLayout organized?
The elements of a BorderLayout are organized at the borders (North, South, East, and West) and the center of a container.

85. What is the % operator?
It is referred to as the modulo or remainder operator. It returns the remainder of dividing the first operand by the second operand.

86. 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.

87. 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.

88. Which class is extended by all other classes?
The Object class is extended by all other classes.

89. Can an object be garbage collected while it is still reachable?
A reachable object cannot be garbage collected. Only unreachable objects may be garbage collected..

90. Is the ternary operator written x : y ? z or x ? y : z ?
It is written x ? y : z.

91. 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.

92. How is rounding performed under integer division?
The fractional part of the result is truncated. This is known as rounding toward zero.

93. 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.

94. 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.

95. 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.

96. 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 access. This means that the class can only be accessed by other classes and interfaces that are defined within the same package.

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

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

99. Does a class inherit the constructors of its superclass?
A class does not inherit constructors from any of its superclasses.

100. For which statements does it make sense to use a label?
The only statements for which it makes sense to use a label are those statements that can enclose a break or continue statement.

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

102. Which TextComponent method is used to set a TextComponent to the read-only state?
setEditable()

103. How are the elements of a CardLayout organized?
The elements of a CardLayout are stacked, one on top of the other, like a deck of cards.

104. Is &&= a valid Java operator?
No, it is not.

105. Name the eight primitive Java types.
The eight primitive types are byte, char, short, int, long, float, double, and boolean.

106. 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.

107. What is the relationship between clipping and repainting?
When a window is repainted by the AWT painting thread, it sets the clipping regions to the area of the window that requires repainting.

108. Is "abc" a primitive value?
The String literal "abc" is not a primitive value. It is a String object.

109. What is the relationship between an event-listener interface and an event-adapter class?
An event-listener interface defines the methods that must be implemented by an event handler for a particular kind of event. An event adapter provides a default implementation of an event-listener interface.

110. What restrictions are placed on the values of each case of a switch statement?
During compilation, the values of each case of a switch statement must evaluate to a value that can be promoted to an int value.

111. What modifiers may be used with an interface declaration?
An interface may be declared as public or abstract.

112. Is a class a subclass of itself?
A class is a subclass of itself.

113. What is the highest-level event class of the event-delegation model?
The java.util.EventObject class is the highest-level class in the event-delegation class hierarchy.

114. What event results from the clicking of a button?
The ActionEvent event is generated as the result of the clicking of a button.

115. 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.

116. What is the difference between a while statement and a do statement?
A while statement checks at the beginning of a loop to see whether the next loop iteration should occur. A do statement checks at the end of a loop to see whether the next iteration of a loop should occur. The do statement will always execute the body of a loop at least once.

117. 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.

118. 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 accomodate platform-specific differences among windowing systems.

119. What is the Collection interface?
The Collection interface provides support for the implementation of a mathematical bag - an unordered collection of objects that may contain duplicates.

120. What modifiers can be used with a local inner class?
A local inner class may be final or abstract.

121. 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.

122. 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.

123. 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.

124. Can an exception be rethrown?
Yes, an exception can be rethrown.

125. Which Math method is used to calculate the absolute value of a number?
The abs() method is used to calculate absolute values.

126. 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.

127. When does the compiler supply a default constructor for a class?
The compiler supplies a default constructor for a class if no other constructors are provided.

128. When is the finally clause of a try-catch-finally statement executed?
The finally clause of the try-catch-finally statement is always executed unless the thread of execution terminates or an exception occurs within the execution of the finally clause.

129. Which class is the immediate superclass of the Container class?
Component

130. 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.

131. How can the Checkbox class be used to create a radio button?
By associating Checkbox objects with a CheckboxGroup.

132. Which non-Unicode letter characters may be used as the first character of an identifier?
The non-Unicode letter characters $ and _ may appear as the first character of an identifier

133. What restrictions are placed on method overloading?
Two methods may not have the same name and argument list but different return types.

134. What happens when you invoke a thread' s interrupt method while it is sleeping or waiting?
When a task' s interrupt() method is executed, the task enters the ready state. The next time the task enters the running state, an InterruptedException is thrown.

135. 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.

136. What is the return type of a program's main() method?
A program's main() method has a void return type.

137. Name four Container classes.
Window, Frame, Dialog, FileDialog, Panel, Applet, or ScrollPane

138. What is the difference between a Choice and a List?
A Choice is displayed in a compact form that requires you to pull it down to see the list of available choices. Only one item may be selected from a Choice. A List may be displayed in such a way that several List items are visible. A List supports the selection of one or more List items.

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

140. What class allows you to read objects directly from a stream?
The ObjectInputStream class supports the reading of objects from input streams.

141. What is the difference between a field variable and a local variable?
A field variable is a variable that is declared as a member of a class. A local variable is a variable that is declared local to a method.

142. 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 dtects that the object has become unreachable.

143. 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.

144. What is the relationship between a method's throws clause and the exceptions
that can be thrown during the method's execution?
A method's throws clause must declare any checked exceptions that are not caught within the body of the method.

145. What is the difference between the JDK 1.02 event model and the event-delegation
model introduced with JDK 1.1?
The JDK 1.02 event model uses an event inheritance or bubbling approach. In this model, components are required to handle their own events. If they do not handle a particular event, the event is inherited by (or bubbled up to) the component's container. The container then either handles the event or it is bubbled up to its container and so on, until the highest-level container has been tried.

In the event-delegation model, specific objects are designated as event handlers for GUI components. These objects implement event-listener interfaces. The event-delegation model is more efficient than the event-inheritance model because it eliminates the processing required to support the bubbling of unhandled events.

146. 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.

147. Why are the methods of the Math class static?
So they can be invoked as if they are a mathematical code library.

148. What Checkbox method allows you to tell if a Checkbox is checked?
getState()

149. What state is a thread in when it is executing?
An executing thread is in the running state.

150. What are the legal operands of the instanceof operator?
The left operand is an object reference or null value and the right operand is a class, interface, or array type.

151. How are the elements of a GridLayout organized?
The elements of a GridBad layout are of equal size and are laid out using the squares of a grid.

152. 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.

153. If an object is garbage collected, can it become reachable again?
Once an object is garbage collected, it ceases to exist. It can no longer become reachable again.

154. 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.

155. What classes of exceptions may be thrown by a throw statement?
A throw statement may throw any expression that may be assigned to the Throwable type.

156. What are E and PI?
E is the base of the natural logarithm and PI is mathematical value pi.

157. Are true and false keywords?
The values true and false are not keywords.

158. What is a void return type?
A void return type indicates that a method does not return a value.

159. 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.

160. 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.

161. What happens when you add a double value to a String?
The result is a String object.

162. What is your platform' s default character encoding?
If you are running Java on English Windows platforms, it is probably Cp1252. If you are running Java on English Solaris platforms, it is most likely 8859_1..

163. Which package is always imported by default?
The java.lang package is always imported by default.

164. 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.

165. How are this and super used?
this is used to refer to the current object instance. super is used to refer to the variables and methods of the superclass of the current object instance.

166. What is the purpose of garbage collection?
The purpose of garbage collection is to identify and discard objects that are no longer needed by a program so that their resources may be reclaimed and reused.

167. What is a compilation unit?
A compilation unit is a Java source code file.

168. What interface is extended by AWT event listeners?
All AWT event listeners extend the java.util.EventListener interface.

169. 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.

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

171. What happens if an exception is not caught?
An uncaught exception results in the uncaughtException() method of the thread's ThreadGroup being invoked, which eventually results in the termination of the program in which it is thrown.

172. What is a layout manager?
A layout manager is an object that is used to organize components in a container.

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

174. 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.

175. Can an abstract class be final?
An abstract class may not be declared as final.

176. 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.

177. What happens if a try-catch-finally statement does not have a catch clause to handle an exception that is thrown within the body of the try statement?
The exception propagates up to the next higher level try-catch statement (if any) or results in the program's termination.

178. What is numeric promotion?
Numeric promotion is the conversion of a smaller numeric type to a larger numeric type, so that integer and floating-point operations may take place. In numerical promotion, byte, char, and short values are converted to int values. The int values are also converted to long values, if necessary. The long and float values are converted to double values, as required.

179. 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.

180. What is the difference between a public and a non-public class?
A public class may be accessed outside of its package. A non-public class may not be accessed outside of its package.

181. To what value is a variable of the boolean type automatically initialized?
The default value of the boolean type is false.

182. Can try statements be nested?
Try statements may be tested.

183. What is the difference between the prefix and postfix forms of the ++ operator?
The prefix form performs the increment operation and returns the value of the increment operation. The postfix form returns the current value all of the expression and then performs the increment operation on that value.

184. What is the purpose of a statement block?
A statement block is used to organize a sequence of statements as a single statement group.

185. 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.

186. What modifiers may be used with a top-level class?
A top-level class may be public, abstract, or final.

187. 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.

188. 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.

189. Can an unreachable object become reachable again?
An unreachable object may become reachable again. This can happen when the object's finalize() method is invoked and the object performs an operation which causes it to become accessible to reachable objects.

190. When is an object subject to garbage collection?
An object is subject to garbage collection when it becomes unreachable to the program in which it is used.

191. What method must be implemented by all threads?

All tasks must implement the run() method, whether they are a subclass of Thread or implement the Runnable interface.

192. What methods are used to get and set the text label displayed by a Button object?
getLabel() and setLabel()

193. Which Component subclass is used for drawing and painting?
Canvas

194. 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.

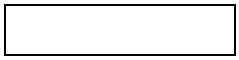
195. What are the two basic ways in which classes that can be run as threads may be defined?
A thread class may be declared as a subclass of Thread, or it may implement the Runnable interface.

196. 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.

197. What is the difference between an if statement and a switch statement?
The if statement is used to select among two alternatives. It uses a boolean expression to decide which alternative should be executed. The switch statement is used to select among multiple alternatives. It uses an int expression to determine which alternative should be executed.

198. What happens when you add a double value to a String?
The result is a String object.

199. What is the List interface?
The List interface provides support for ordered collections of objects.



Database - JDBC (java.sql)

Connecting to a Database

This example uses the JDBC-ODBC bridge to connect to a database called "mydatabase".

```
try {
  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

  String url = "jdbc:odbc:mydatabase";
  Connection con = DriverManager.getConnection(
    url, "login", "password");
} catch (ClassNotFoundException e) {
} catch (SQLException e) {
}
```

Creating a Table

This example creates a table called "mytable" with three columns: COL_A which holds strings, COL_B which holds integers, and COL_C which holds floating point numbers.

```
try {
  Statement stmt = con.createStatement();
```

```
  stmt.executeUpdate("CREATE TABLE mytable(
    COL_A VARCHAR(100), COL_B INTEGER, COL_C FLOAT)");
} catch (SQLException e) {
}
```

Entering a New Row into a Table

This example enters a row containing a string, an integer, and a floating point number into the table called ''mytable''.

```
try {
  Statement stmt = connection.createStatement();
  stmt.executeUpdate("INSERT INTO mytable
    VALUES ('Patrick Chan', 123, 1.23)");
  connection.close();
} catch (SQLException e) {
}
```

Getting All Rows from a Table

This example retrieves all the rows from a table called ''mytable''. A row in ''mytable'' consists of a string, integer, and floating point number.

```
try {
  Statement stmt = connection.createStatement();

  // Get data using column names.
  ResultSet rs = stmt.executeQuery(
    "SELECT * FROM mytable");
  while (rs.next()) {
    String s = rs.getString("COL_A");
    int i = rs.getInt("COL_B");
    float f = rs.getFloat("COL_C");
    process(s, i, f);
  }

  // Get data using column numbers.
  rs = stmt.executeQuery(
    "SELECT * FROM mytable");
  while (rs.next()) {
    String s = rs.getString(1);
    int i = rs.getInt(2);
    float f = rs.getFloat(3);
    process(s, i, f);
  }
} catch (SQLException e) {
}
```

Getting Particular Rows from a Table

This example retrieves all rows from a table called ''mytable'' whose column COL_A equals ``Patrick Chan''. A row in ''mytable'' consists of a string, integer, and floating point number.

```
try {
  Statement stmt = connection.createStatement();
  ResultSet rs = stmt.executeQuery(
    "SELECT * FROM mytable WHERE COL_A = 'Patrick Chan'");
  rs.next();
  String s = rs.getString("COL_A");
  int i = rs.getInt("COL_B");
  float f = rs.getFloat("COL_C");
  process(s, i, f);
```

```
} catch (SQLException e) {
}
```

Updating a Row of Data in a Table

This example updates a row in a table called ``mytable''. In particular, for all rows whose column COL_B equals 123, column COL_A is set to ''John Doe''.

```
try {
  Statement stmt = connection.createStatement();
  int numUpdated = stmt.executeUpdate(
    "UPDATE mytable SET COL_A = 'John Doe'
    WHERE COL_B = 123");
  connection.close();
} catch (SQLException e) {
}
```

Using a Prepared Statement

A prepared statement should be used in cases where a particular SQL statement is used frequently. The prepared statement is more expensive to set up but executes faster than a statement. This example demonstrates a prepared statement for getting all rows from a table called ''mytable'' whose column COL_A equals ''Patrick Chan''. This example also demonstrates a prepared statement for updating data in the table. In particular, for all rows whose column COL_B equals 123, column COL_A is set to ''John Doe''.

```
try {
  // Retrieving rows from the database.
  PreparedStatement stmt = connection.prepareStatement(
    "SELECT * FROM mytable WHERE COL_A = ?");
  int column = 1;
  stmt.setString(column, "Patrick Chan");
  ResultSet rs = stmt.executeQuery();

  // Updating the database.
  stmt = connection.prepareStatement(
    "UPDATE mytable SET COL_A = ? WHERE COL_B = ?");
  column = 1;
  stmt.setString(column, "John Doe");
  column = 2;
  stmt.setInt(column, 123);
  int numUpdated = stmt.executeUpdate();
} catch (SQLException e) {
}
```


Handling Events with an Anonymous Class

If an event handler is specific to a component (that is, not shared by other components), there is no need to declare a class to handle the event. The event handler can be implemented using an anonymous inner class. This example demonstrates an anonymous inner class to handle key events for a component.

```
component.addKeyListener(new KeyAdapter() {
  public void keyPressed(KeyEvent evt) {
  }
});
```

Handling Action Events

Action events are fired by subclasses of AbstractButton and includes buttons, checkboxes, and menus.

```
AbstractButton button = new JButton(quot;OK");
button.addActionListener(new MyActionListener());

public class MyActionListener
  implements ActionListener {
    public void actionPerformed(ActionEvent evt) {
      // Determine which abstract
      // button fired the event.
      AbstractButton button =
        (AbstractButton)evt.getSource();
    }
}
```

Handling Key Presses

You can get the key that was pressed either as a key character (which is a Unicode character) or as a key code (a special value representing a particular key on the keyboard).

```
component.addKeyListener(new MyKeyListener());

public class MyKeyListener extends KeyAdapter {
  public void keyPressed(KeyEvent evt) {
    // Check for key characters.
    if (evt.getKeyChar() == ') {
      process(evt.getKeyChar());
    }

    // Check for key codes.
    if (evt.getKeyCode() == KeyEvent.VK_HOME) {
      process(evt.getKeyCode());
    }
  }
}
```

Handling Mouse Clicks

```
component.addMouseListener(
  new MyMouseListener());

public class MyMouseListener
  extends MouseAdapter {
    public void mouseClicked(MouseEvent evt) {
      if ((evt.getModifiers() &
        InputEvent.BUTTON1_MASK) != 0) {
          processLeft(evt.getPoint());
      }
      if ((evt.getModifiers() &
        InputEvent.BUTTON2_MASK) != 0) {
          processMiddle(evt.getPoint());
      }
      if ((evt.getModifiers() &
        InputEvent.BUTTON3_MASK) != 0) {
          processRight(evt.getPoint());
      }
    }
}
```

Handling Mouse Motion

```
component.addMouseMotionListener(
```

```
  new MyMouseMotionListener());

    public class MyMouseMotionListener
      extends MouseMotionAdapter {
        public void mouseMoved(MouseEvent evt) {
          // Process current position of cursor
          // while all mouse buttons are up.
          process(evt.getPoint());
        }
        public void mouseDragged(MouseEvent evt) {
          // Process current position of cursor
          // while mouse button is pressed.
          process(evt.getPoint());
        }
    }
```

Detecting Double and Triple Clicks

```
component.addMouseListener(
  new MyMouseListener());

public class MyMouseListener extends MouseAdapter {
  public void mouseClicked(MouseEvent evt) {
    if (evt.getClickCount() == 3) {
      // triple-click
    } else if (evt.getClickCount() == 2) {
      // double-click
    }
  }
}
```

Handling Focus Changes

```
component.addFocusListener(
  new MyFocusListener());

public class MyFocusListener
  extends FocusAdapter {
    public void focusGained(FocusEvent evt) {
      // The component gained the focus.
    }
    public void focusLost(FocusEvent evt) {
      // The component lost the focus.
    }
}
```


Constructing a Path

On Windows, this example creates the path \blash a\blash b. On Unix, the path would be /a/b.

```
String path = File.separator +
  "a" + File.separator + "b";
```

Reading Text from Standard Input

```
try {
  BufferedReader in = new BufferedReader(
    new InputStreamReader(System.in));
```

```
  String str = "";
  while (str != null) {
    System.out.print("> prompt ");
    str = in.readLine();
    process(str);
  }
} catch (IOException e) {
}
```

Reading Text from a File

```
try {
  BufferedReader in = new BufferedReader(
    new FileReader("infilename"));
  String str;
  while ((str = in.readLine()) != null) {
    process(str);
  }
  in.close();
} catch (IOException e) {
}
```

Writing to a File

If the file does not already exist, it is automatically created.

```
try {
  BufferedWriter out = new BufferedWriter(
    new FileWriter("outfilename"));
  out.write("aString");
  out.close();
} catch (IOException e) {
}
```

Creating a Directory

```
(new File("directoryName")).mkdir();
```

Appending to a File

```
try {
  BufferedWriter out = new BufferedWriter(
    new FileWriter("filename", true));
  out.write("aString");
  out.close();
} catch (IOException e) {
}
```

Deleting a File

```
(new File("filename")).delete();
```

Deleting a Directory

```
(new File("directoryName")).delete();
```

Creating a Temporary File

```
try {
  // Create temp file.
```

```
  File temp = File.createTempFile(
    "pattern", ".suffix");

  // Delete temp file when program exits.
  temp.deleteOnExit();

  // Write to temp file
  BufferedWriter out = new BufferedWriter(
    new FileWriter(temp));
  out.write("aString");
  out.close();
  } catch (IOException e) {
}
```

Using a Random Access File

```
try {
  File f = new File("filename");
  RandomAccessFile raf =
    new RandomAccessFile(f, "rw");

  // Read a character.
  char ch = raf.readChar();

  // Seek to end of file.
  raf.seek(f.length());

  // Append to the end.
  raf.writeChars("aString");
  raf.close();
} catch (IOException e) {
}
```

Serializing an Object

The object to be serialized must implement java.io.Serializable.

```
try {
  ObjectOutput out = new ObjectOutputStream(
    new FileOutputStream("filename.ser"));
  out.writeObject(object);
  out.close();
} catch (IOException e) {
}
```

Deserializing an Object

This example deserializes a java.awt.Button object.

```
try {
  ObjectInputStream in = new ObjectInputStream(
    new FileInputStream("filename.ser"));
  AnObject object = (AnObject) in.readObject();
  in.close();
} catch (ClassNotFoundException e) {
} catch (IOException e) {
}
```

Traversing a Directory

```
public static void traverse(File f) {
```

```
  process(f);

  if (f.isDirectory()) {
    String[] children = f.list();
    for (int i=0; i<children.length; i++) {
      traverse(new File(f, children[i]));
    }
  }
}
```

Reading UTF-8 Encoded Data

```
try {
  BufferedReader in = new BufferedReader(
    new InputStreamReader(new FileInputStream(
    "infilename"), "UTF8"));
  String str = in.readLine();
} catch (UnsupportedEncodingException e) {
} catch (IOException e) {
}
```

Writing UTF-8 Encoded Data

```
try {
  Writer out = new BufferedWriter(
    new OutputStreamWriter(new FileOutputStream(
    "outfilename"), "UTF8"));
  out.write(aString);
  out.close();
} catch (UnsupportedEncodingException e) {
} catch (IOException e) {
}
```

Reading ISO Latin-1 Encoded Data

```
try {
  BufferedReader in = new BufferedReader(
    new InputStreamReader(new FileInputStream(
    "infilename"), "8859_1"));
  String str = in.readLine();
} catch (UnsupportedEncodingException e) {
} catch (IOException e) {
}
```

Writing ISO Latin-1 Encoded Data

```
try {
  Writer out = new BufferedWriter(
    new OutputStreamWriter(new FileOutputStream(
    "outfilename"), "8859_1"));
  out.write(aString);
  out.close();
} catch (UnsupportedEncodingException e) {
} catch (IOException e) {
}
```

Creating a URL

```
try {
 // With components.
   URL url = new URL("http","hostname", 80, "index.html");

// With a single string.
  url = new URL(
  "http://hostname:80/index.html");
   } catch (MalformedURLException e) {
   }
```

Parsing a URL

```
try {
 URL url = new URL("http://hostname:80/index.html#_top_");
 String protocol = url.getProtocol();    // http
 String host = url.getHost();  // hostname
  int port = url.getPort();    // 80
  String file = url.getFile(); // index.html
  String ref = url.getRef();    // _top_
   } catch (MalformedURLException e) {
   }
```

Reading Text from a URL

```
try {
 URL url = new URL("http://hostname:80/index.html");
 BufferedReader in = new BufferedReader(
       new InputStreamReader(url.openStream()));

 String str;
  while ((str = in.readLine()) != null) {
 process(str);
    }
   in.close();
   } catch (MalformedURLException e) {
   } catch (IOException e) {
   }
```

Resolving a Hostname

Creating a Client Socket

```
try {
 InetAddress addr = InetAddress.getByName("java.sun.com");
 int port = 80;
 Socket sock = new Socket(addr, port);
  } catch (IOException e) {
  }
```

Creating a Server Socket

```
try {
 int port = 2000;
  ServerSocket srv = new ServerSocket(port);

 // Wait for connection from client.
```

```
 Socket socket = srv.accept();
 } catch (IOException e) {
 }
```

Reading Text from a Socket

```
try {
 BufferedReader rd = new BufferedReader(
 new InputStreamReader(socket.getInputStream()));

 String str;
 while ((str = rd.readLine()) != null) {
 process(str);

 }
     rd.close();
  } catch (IOException e) {
  }
```

Writing Text to a Socket

```
try {
 BufferedWriter wr = new BufferedWriter(
  new OutputStreamWriter(socket.getOutputStream()));
    wr.write("aString");
     wr.flush();
  } catch (IOException e) {
  }
```

Sending a Datagram

```
public static void send(InetAddress dst,
          int port, byte[] outbuf, int len) {
try {
 DatagramPacket request = new DatagramPacket(
                  outbuf, len, dst, port);
  DatagramSocket socket = new DatagramSocket();
    socket.send(request);
  } catch (SocketException e) {
  } catch (IOException e) {
  }
 }
```

Receiving a Datagram

```
try {
 byte[] inbuf = new byte[256]; // default size
 DatagramSocket socket = new DatagramSocket();

 // Wait for packet
 DatagramPacket packet = new DatagramPacket(
                    inbuf, inbuf.length);
  socket.receive(packet);

 // Data is now in inbuf
 int numBytesReceived = packet.getLength();
 } catch (SocketException e) {
 } catch (IOException e) {
 }
```

Joining a Multicast Group

```
public void join(String groupName, int port) {
 try {
  MulticastSocket msocket = new MulticastSocket(port);
  group = InetAddress.getByName(groupName);
   msocket.joinGroup(group);
   } catch (IOException e) {
    }
   }
```

Receiving from a Multicast Group

```
public void read(MulticastSocket msocket,
                 byte[] inbuf) {
try {
 DatagramPacket packet = new DatagramPacket(
                  inbuf, inbuf.length);

 // Wait for packet
 msocket.receive(packet);

// Data is now in inbuf
int numBytesReceived = packet.getLength();
   } catch (IOException e) {
    }
   }
```

Sending to a Multicast Group

```
 byte[] outbuf = new byte[1024];
 int port = 1234;
  try {
  DatagramSocket socket = new DatagramSocket();
  InetAddress groupAddr = InetAddress.getByName(
  "228.1.2.3");
   DatagramPacket packet = new DatagramPacket(
         outbuf, outbuf.length, groupAddr, port);
   socket.send(packet);
   } catch (SocketException e) {
   } catch (IOException e) {
   }
```

Defining and Exporting a Remote Object

1. Define the remote interface.

```
   import java.rmi.*;

 public interface RObject extends Remote {
  void aMethod() throws RemoteException;
    }
```

Looking Up a Remote Object and Invoking a Method

```
 try {
 // Look up remote object
 RObject robj = (RObject) Naming.lookup(
 "//localhost/RObjectServer");
```

```
// Invoke method on remote object
 robj.aMethod();
} catch (MalformedURLException e) {
} catch (UnknownHostException e) {
} catch (NotBoundException e) {
} catch (RemoteException e) {
}
```

Passing Parameters to a Remote Method

Arguments to remote methods must be primitive, serializable, or Remote. This example demonstrates the declaration and use of all three parameter types.

1. Define the remote interface.

```
import java.rmi.*;

public interface RObject extends Remote {
 // This parameter is primitive.
 void primitiveArg(int num) throws RemoteException;

// This parameter implements Serializable.
  void byValueArg(Integer num) throws RemoteException;

// This parameter implements Remote.
  void byRefArg(ArgObject arg) throws RemoteException;
  }

  public interface ArgObject extends Remote {
     int aMethod() throws RemoteException;
  }
```

2. Define the remote object implementation.

```
import java.rmi.*;
import java.rmi.server.UnicastRemoteObject;

public class RObjectImpl extends UnicastRemoteObject implements RObject {
 public RObjectImpl() throws RemoteException {
      super();
  }
 public void primitiveArg(int num) throws RemoteException {
  }
 public void byValueArg(Integer num) throws RemoteException {
  }
 public void byRefArg(ArgObject arg) throws RemoteException {
  }
  }
```

3. Compile the remote object implementation.

```
  > javac RObject.java RObjectImpl.java
```

4. Generate the skeletons and stubs.

```
  > rmic RObjectImpl
```

5. Create an instance of RObjectImpl and bind it to the RMI Registry.

```
  try {
     RObject robj = new RObjectImpl();
```

```
        Naming.rebind("//localhost/RObjectServer", robj);
    } catch (MalformedURLException e) {
    } catch (UnknownHostException e) {
    } catch (RemoteException e) {
    }
```

6. Look Up the Remote object and pass the parameters.

```
    try {
        // Look up the remote object
        RObject robj = (RObject) Naming.lookup("//localhost/RObjectServer");

        // Pass a primitive value as argument
        robj.primitiveArg(1998);

        // Pass a serializable object as argument
        robj.byValueArg(new Integer(9));

        // Pass a Remote object as argument
        robj.byRefArg(new ArgObjectImpl());
    } catch (MalformedURLException e) {
    } catch (UnknownHostException e) {
    } catch (NotBoundException e) {
    } catch (RemoteException e) {
    }
```

Returning Values from a Remote Method

Return values from remote methods must be primitive, serializable, or Remote. This example demonstrates the declaration and use of all three return types. 1. Define the remote interface.

```
import java.rmi.*;
 public interface RObject extends Remote {
 // This return value is primitive.
  int primitiveRet() throws RemoteException;

 // This return value implements Serializable.
   Integer byValueRet() throws RemoteException;

 // This return value implements Remote.
     ArgObject byRefRet() throws RemoteException;
    }

    public interface ArgObject extends Remote {
        int aMethod() throws RemoteException;
    }
```

2. Define the remote object implementation.

```
import java.rmi.*;
import java.rmi.server.UnicastRemoteObject;

public class RObjectImpl extends UnicastRemoteObject
                         implements RObject {
  public RObjectImpl() throws RemoteException {
          super();
     }
 public int primitiveRet() throws RemoteException {
      return 3000;
    }
```

```
 public Integer byValueRet() throws RemoteException {
  return new Integer(2000);
  }
  public ArgObject byRefRet() throws RemoteException {
   return new ArgObjectImpl();
  }
 }
```

3. Compile the remote object implementation.
   > javac RObject.java RObjectImpl.java

4. Generate the skeletons and stubs.
 > rmic RObjectImpl

5. Create an instance of RObjectImpl and bind it to the RMI Registry.

```
 try {
RObject robj = new RObjectImpl();
   Naming.rebind("//localhost/RObjectServer", robj);
  } catch (MalformedURLException e) {
 } catch (UnknownHostException e) {
  } catch (RemoteException e) {
 }
```

6. Look Up the Remote object, invoke the methods, and receive the return values.

```
 try {
 // Look up the remote object
 RObject robj = (RObject) Naming.lookup(
                    "//localhost/RObjectServer");

 // Receive the primitive value as return value
 int r1 = robj.primitiveRet();

 // Receive the serializable object as return value
 Integer r2 = robj.byValueRet();

  // Receive the Remote Object as return value
   ArgObject aobj =  robj.byRefRet();
 } catch (MalformedURLException e) {
 } catch (UnknownHostException e) {
   } catch (NotBoundException e) {
  } catch (RemoteException e) {
   }
```

Throwing an Exception from a Remote Method

1. Define the remote interface.

```
 import java.rmi.*;

 public interface RObject extends Remote {
   void aMethod() throws RemoteException;
    }
```

2. Define the remote object implementation.

```
import java.rmi.*;
import java.rmi.server.UnicastRemoteObject;

public class RObjectImpl extends
```

```
 UnicastRemoteObject implements RObject {
 public RObjectImpl() throws RemoteException {
   super();
 }
public void doMethod() throws RemoteException {
// The actual exception must be wrapped in
// a RemoteException
   throw new RemoteException(
 "message", new FileNotFoundException("message"));
    }
   }
```

3. Compile the remote object implementation.

```
> javac RObject.java RObjectImpl.java
```

4. Generate the skeletons and stubs.

```
> rmic RObjectImpl
```

5. Create an instance of RObjectImpl and bind it to the RMI Registry.

```
 try {
 RObject robj = new RObjectImpl();
  Naming.rebind("//localhost/RObjectServer", robj);
   } catch (MalformedURLException e) {
   } catch (UnknownHostException e) {
   } catch (RemoteException e) {
   }
```

6. Look up the Remote object, invoke the method, and catch the exception.

```
 try {
 // Look up the remote object.
 RObject robj = (RObject) Naming.lookup(
 "//localhost/RObjectServer");

 // Invoke the method.
  robj.doMethod();
   } catch (MalformedURLException e) {
   } catch (UnknownHostException e) {
   } catch (NotBoundException e) {
   } catch (RemoteException e) {
 // Get the actual exception that was thrown.
   Throwable realException = e.detail;
   }
```


Constructing a String

If you are constructing a string with several appends, it may be more efficient to construct it using a StringBuffer and then convert it to an immutable String object.

```
StringBuffer buf = new StringBuffer("Initial Text");

// Modify
int index = 1;
buf.insert(index, "abc");
buf.append("def");

 // Convert to string
```

```
String s = buf.toString();
```

Getting a Substring from a String

```
int start = 1;
int end = 4;
String substr = "aString".substring(start, end);   // Str
```

Searching a String

```
String string = "aString";

// First occurrence.
int index = string.indexOf('S'); // 1

// Last occurrence.
index = string.lastIndexOf('r'); // 4

// Not found.
index = string.lastIndexOf('z'); // -1
```

Replacing Characters in a String

```
// Replace all occurrences of 'a' with 'o'
String newString = string.replace('a', 'o');
```

Replacing Substrings in a String

```
static String replace(String str,
        String pattern, String replace) {
  int s = 0;
  int e = 0;
StringBuffer result = new StringBuffer();

 while ((e = str.indexOf(pattern, s)) >= 0) {
   result.append(str.substring(s, e));
   result.append(replace);
   s = e+ pattern.length();
    }
   result.append(str.substring(s));
   return result.toString();
   }
```

Converting a String to Upper or Lower Case

```
// Convert to upper case
String upper = string.toUpperCase();

// Convert to lower case
String lower = string.toLowerCase();
```

Converting a String to a Number

```
int i = Integer.parseInt("123");
long l = Long.parseLong("123");
float f = Float.parseFloat("123.4");
double d = Double.parseDouble("123.4e10");
```

Converting Unicode to UTF-8

```
try {
 String string = "\u5639\u563b";
 byte[] utf8 = string.getBytes("UTF8");
 } catch (UnsupportedEncodingException e) {
    }
```

Converting UTF-8 to Unicode

```
 public static String toUnicode(byte[] utf8buf) {
  try {
    return new String(utf8buf, "UTF8");
    } catch (UnsupportedEncodingException e) {
    }
    return null;
    }
```

Determining a Character's Unicode Block

```
char ch = '\u5639';
Character.UnicodeBlock block =
          Character.UnicodeBlock.of(ch);
```

Breaking a String into Words

```
String aString = "word1 word2 word3";
StringTokenizer parser =
         new StringTokenizer(aString);
while (parser.hasMoreTokens()) {
        processWord(parser.nextToken());
    }
```

What Is an Enterprise Bean?

Written in the Java programming language, an *enterprise bean* is a server-side component that encapsulates the business logic of an application. The business logic is the code that fulfills the purpose of the application. In an inventory control application, for example, the enterprise beans might implement the business logic in methods called checkInventoryLevel and orderProduct. By invoking these methods, remote clients can access the inventory services provided by the application.

Benefits of Enterprise Beans

For several reasons, enterprise beans simplify the development of large, distributed applications. First, because the EJB container provides system-level services to enterprise beans, the bean developer can concentrate on solving business problems. The EJB container--not the bean developer--is responsible for system-level services such as transaction management and security authorization.

Second, because the beans--and not the clients--contain the application' s business logic, the client developer can focus on the presentation of the client. The client developer does not have to code the routines that implement business rules or access databases. As a result, the clients are thinner, a benefit that is particularly important for clients that run on small devices.

Third, because enterprise beans are portable components, the application assembler can build new applications from existing beans. These applications can run on any compliant J2EE server.

When to Use Enterprise Beans

You should consider using enterprise beans if your application has any of the following requirements:

The application must be scalable. To accommodate a growing number of users, you may need to distribute an application's components across multiple machines. Not only can the enterprise beans of an application run on different machines, but their location will remain transparent to the clients.

Transactions are required to ensure data integrity. Enterprise beans support transactions, the mechanisms that manage the concurrent access of shared objects.

The application will have a variety of clients. With just a few lines of code, remote clients can easily locate enterprise beans. These clients can be thin, various, and numerous.

Types of Enterprise Beans

Table 3-1 summarizes the three different types of enterprise beans. The following sections discuss each type in more detail.

Table 3-1 Summary of Enterprise Bean Types

Enterprise Bean Type	Purpose
Session	Performs a task for a client
Entity	Represents a business entity object that exists in persistent storage
Message-Driven	Acts as a listener for the Java Message Service API, processing messages asynchronously

1. What is Entity Bean and Session Bean ?

A *session bean* represents a single client inside the J2EE server. To access an application that is deployed on the server, the client invokes the session bean's methods. The session bean performs work for its client, shielding the client from complexity by executing business tasks inside the server.

As its name suggests, a session bean is similar to an interactive session. A session bean is not shared--it may have just one client, in the same way that an interactive session may have just one user. Like an interactive session, a session bean is not persistent. (That is, its data is not saved to a database.) When the client terminates, its session bean appears to terminate and is no longer associated with the client.

State Management Modes

There are two types of session beans: stateful and stateless.

Stateful Session Beans

The state of an object consists of the values of its instance variables. In a stateful session bean, the instance variables represent the state of a unique client-bean session. Because the client interacts ("talks") with its bean, this state is often called the *conversational state*.

The state is retained for the duration of the client-bean session. If the client removes the bean or terminates, the session ends and the state disappears. This transient nature of the state is not a problem, however, because when the conversation between the client and the bean ends there is no need to retain the state.

Stateless Session Beans

A stateless session bean does not maintain a conversational state for a particular client. When a client invokes the method of a stateless bean, the bean's instance variables may contain a state, but only for the duration of the invocation. When the method is finished, the state is no longer retained. Except during method invocation, all instances of a stateless bean are equivalent, allowing the EJB container to assign an instance to any client.

Because stateless session beans can support multiple clients, they can offer better scalability for applications that require large numbers of clients. Typically, an application requires fewer stateless session beans than stateful session beans to support the same number of clients.

At times, the EJB container may write a stateful session bean to secondary storage. However, stateless session beans are never written to secondary storage. Therefore, stateless beans may offer better performance than stateful beans.

A stateless session bean is an enterprise bean (EJB component) that provides a stateless service to the client. Conceptually, the business methods on a stateless session bean are similar to procedural applications or static methods; there is no instance state, so all the data needed to execute the method is provided by the method arguments.

The stateless session bean is an EJB component that implements the `javax.ejb.SessionBean` interface and is deployed with the declarative attribute "stateless". Stateless session beans are called "stateless" because they do not maintain conversational state specific to a client session. In other words, the instance fields in a stateless session bean do not maintain data relative to a client session. This makes stateless session beans very lightweight and fast, but also limits their behavior.

Stateless session beans are usually developed as a set of related and simple services. Think of a session bean as a functional API where each business method represents an isolated independent service. An example is a CreditService bean that provides methods for making charges against different types of credit cards (MC, Visa, Discovery, etc).

```
public class CreditService implements javax.ejb.SessionBean {

   public Charge charge (String accountToCredit, CreditCard card, double amount)
   throws ValidationException, RemoteException{

         // attempt to obtain Merchant bean representing the retailer making the charge.
         try{
            MerchantHome mrchntHome =  ... get VendorHome reference
            Merchant merchant =  mrchntHome.findByPrimaryKey(vendorToCredit);
         }catch(ObjectNotFoundException onfe){
               throw new ValidationException("Invalid merchant account
number");
```

```
        }

        // attempt to create a Charge bean based on the vendor, amount, and credit card
info
        try{
           ChargeHome chgHome = ... get ChargeHome reference
           Charge charge =  chgHome.create(merchant, card, amount);
            return charge;
        }catch(CreateException ce){
              throw new ValidationException(ce.getMessage());
        }
  }
...
}
```

In the above example the `CreditService` bean method `charge( )` is completely independent of the bean state or any other methods. The `charge( )` method is a stateless service and the `CreditService` bean is a stateless bean. The `charge( )` method uses two other bean types to process the charge. Its normal for stateless beans to use other beans but they can also access the database directly or even other resources like proprietary connections. Below is a list the beans used in this example.

Bean Name	**Bean Type**
CreditService	Stateless Session
Merchant	Entity
Charge	Entity

When to Use Session Beans

In general, you should use a session bean if the following circumstances hold:
At any given time, only one client has access to the bean instance.
The state of the bean is not persistent, existing only for a short period of time (perhaps a few hours).

Stateful session beans are appropriate if any of the following conditions are true:

A stateful session bean is an enterprise bean (EJB component) that acts as a server-side extension of the client that uses it. The stateful session bean is created by a client and will work for only that client until the client connection is dropped or the bean is explicitly removed.

The stateful session bean is EJB component that implements the javax.ejb.SessionBean interface and is deployed with the declarative attribute "stateful". Stateful session beans are called "stateful" because they maintain a conversational state with the client. In other words, they have state or instance fields that can be initialized and changed by the client with each method invocation. The bean can use the conversational state as it process business methods invoked by the client.

Stateful session beans are usually developed to act as agents for the client, managing the interaction of other beans and performing work on behalf of the client application. An example is a shopping cart stateful session bean that tracks a client's product choices and can execute a sale when requested. Below is partial example of a stateful session bean that is used as a shopping cart.

In the example above the `ShoppingCartBean` keeps track of the `Customer` and the items chosen by the client application. This is the bean's conversational state. Once the client application is ready to execute a sale, the `ShoppingCartBean` manages the interactions of the `Customer, Product, CreditServer,` and `Order` beans in a workflow that results in a charge against the customers credit card and the creation of a shipping order. The `ShoppingCartBean` behaves as an agent for the client managing the interaction of these other beans, tracking chosen items, and executing the sale. The below table identifies the beans types used in the `ShoppingCartBean` example above.

Bean Name	**Bean Type**
ShoppingCartBean	Stateful Session
CreditService	Stateless Session
Customer	Entity
Product	Entity
Order	Entity

From the clients perspective only the `Customer` and `ShoppingCart` beans are visible, because the client application works with the bean's remote interface while the bean itself resides on the server. This means that the client need only be concerned with the business methods made public by the `ShoppingCartBean`. Below is an example of the client applications view of the `ShoppingCartBean`. (Note: The client interacts with the beans remote interface not the bean itself. The remote interface is called the `ShoppingCart`.) Session beans like the `shoppingCartBean` only live as long as the client maintains a connection. In other words, they represent some aspect of the client's current session with the system and die when the client ends that session. Session beans are generally not fault tolerant. A system failure or shut down will result in the death of session bean and a loss of any conversation state it maintained prior to the failure. (This is a conceptual lifecycle of a session bean. Some vendors can make them more fault tolerant and longer lasting).

The bean's state represents the interaction between the bean and a specific client.
The bean needs to hold information about the client across method invocations.
The bean mediates between the client and the other components of the application, presenting a simplified view to the client.
Behind the scenes, the bean manages the work flow of several enterprise beans. For an example, see the AccountControllerEJB session bean in Chapter 18.
To improve performance, you might choose a stateless session bean if it has any of these traits:
The bean's state has no data for a specific client.
In a single method invocation, the bean performs a generic task for all clients. For example, you might use a stateless session bean to send an e-mail that confirms an online order.
The bean fetches from a database a set of read-only data that is often used by clients. Such a bean, for example, could retrieve the table rows that represent the products that are on sale this month.

What Is an Entity Bean?
An *entity bean* represents a business object in a persistent storage mechanism. Some examples of business objects are customers, orders, and products. In the J2EE SDK, the persistent

storage mechanism is a relational database. Typically, each entity bean has an underlying table in a relational database, and each instance of the bean corresponds to a row in that table. For code examples of entity beans, please refer to chapters 5 and 6.

What Makes Entity Beans Different from Session Beans?

Entity beans differ from session beans in several ways. Entity beans are persistent, allow shared access, have primary keys, and may participate in relationships with other entity beans.

Persistence

Because the state of an entity bean is saved in a storage mechanism, it is persistent. *Persistence* means that the entity bean's state exists beyond the lifetime of the application or the J2EE server process. If you've worked with databases, you're familiar with persistent data. The data in a database is persistent because it still exists even after you shut down the database server or the applications it services.

There are two types of persistence for entity beans: bean-managed and container-managed. With *bean-managed persistence*, the entity bean code that you write contains the calls that access the database. If your bean has container-managed persistence, the EJB container automatically generates the necessary database access calls. The code that you write for the entity bean does not include these calls. For additional information, see the section Container-Managed Persistence.

Shared Access

Entity beans may be shared by multiple clients. Because the clients might want to change the same data, it's important that entity beans work within transactions. Typically, the EJB container provides transaction management. In this case, you specify the transaction attributes in the bean's deployment descriptor. You do not have to code the transaction boundaries in the bean--the container marks the boundaries for you. See Chapter 14 for more information.

Primary Key

Each entity bean has a unique object identifier. A customer entity bean, for example, might be identified by a customer number. The unique identifier, or *primary key*, enables the client to locate a particular entity bean. For more information see the section Primary Keys for Bean-Managed Persistence.

Relationships

Like a table in a relational database, an entity bean may be related to other entity beans. For example, in a college enrollment application, `StudentEJB` and `CourseEJB` would be related because students enroll in classes.

You implement relationships differently for entity beans with bean-managed persistence and those with container-managed persistence. With bean-managed persistence, the code that you write implements the relationships. But with container-managed persistence, the EJB container takes care of the relationships for you. For this reason, relationships in entity beans with container-managed persistence are often referred to as *container-managed relationships*.

Container-Managed Persistence

The term *container-managed persistence* means that the EJB container handles all database access required by the entity bean. The bean's code contains no database access (SQL) calls. As a result, the bean's code is not tied to a specific persistent storage mechanism (database). Because of this flexibility, even if you redeploy the same entity bean on different J2EE servers that use different databases, you won't need to modify or recompile the bean's code. In short, your entity beans are more portable.

In order to generate the data access calls, the container needs information that you provide in the entity bean's abstract schema.

When to Use Entity Beans

You should probably use an entity bean under the following conditions:

The bean represents a business entity, not a procedure. For example, `CreditCardEJB` would be an entity bean, but `CreditCardVerifierEJB` would be a session bean.

The bean's state must be persistent. If the bean instance terminates or if the J2EE server is shut down, the bean's state still exists in persistent storage (a database).

1. What Is an Entity Bean?

An entity bean is a type of enterprise bean; a type of EJB server-side component. Entity bean components implement the `javax.ejb.EntityBean` interface and can be container-managed (CMP) or bean-managed (BMP). Entity beans are designed to represent data in the database; they wrapper data with business object semantics and read and update data automatically.

Entity beans have identity, where each bean represents a unique set of data. An entity bean of type Customer, for example, would have thousands of identities -- one for each customer in the database. Entity beans can be safely shared by many clients, so that several applications can access the same bean identity at the concurrently.

A bean developer might choose to create a Customer bean to represent customer data in the database. The bean developer decides what data from the database represents a customer and how customers are uniquely identified. The bean developer also decides what business methods the Customer bean expose and what data can be changed or read. Below is the server-side component definition of a Customer bean.

```
public class CustomerBean implements javax.ejb.EntityBean {

    // persistent fields
    public long customerNumber;
    public Date lastActive;
    public String lastName;
    public String firstName;

    // initialization method for creating a new customer
    public CustomerKey ejbCreate(long id, PersonName name){
        customerNumber = id;
        setName(name);
        lastActive = new Date();
        return new CustomerKey(id);
    }

    // business methods
    public PersonName getName( ){
        return new PersonName(firstName, lastName);
    }
    public void setName(PersonName name){
        lastName = name.getLastName();
        fistName = name.getFirstName();
```

```
    }
    public Date lastActive( ){
        return lastActive;
    }
    ...
}
```

Note: Only a subset of the bean code normally implemented is shown here for brevity. Ordinarily the bean would also implement special EJB specific callback methods that are not important to this FAQ entry.

The `ejbCreate( )` method is used only once in the life type of a bean identity when its first created. In this case we initialize the bean with the customer's id number and name. The `lastActive` value is set within the method. The `ejbCreate( )` creates and returns the customer bean's identity, which is called a *primary key* in EJB. A primary key is a object that represents a unique identifier for a bean. In the case of the Customer bean the primary key is the `CustomerKey` which is an object that wrappers the customer's id. Primary keys can be as simple as a String value or more be more complicated like a key that wraps several values that together represent a unique index to the bean identity. Below is the definition of the `CustomerKey` object.

```
public class CustomerKey implements java.io.Serializable {
    public long id;

    public CustomerKey(long l)
    {
        id = l;
    }
    public int hashCode( ) {
        return (int)id;
    }
    public boolean equals(Object otherKey){
        if(otherKey instanceof CustomerKey)
            return ((CustomerKey)otherKey).id == this.id;
        else
            return false;
    }
}
```

You will notice that the business methods do not exactly match the persistent fields. This illustrates that the persistent state of the bean need not map exactly to the business methods, which provides more flexibility in how components are designed. For example, instead of having an accessor for `lastName` and `firstName`, the Customer bean uses a serializable `PersonName` to pass this information to and from the bean's client. In addition, you will have noticed that the `lastActive` can only be read by a bean client. The bean itself is responsible for updating this value.

In addition to the business methods and the `ejbCreate( )` methods there are a number of notification methods that the bean container (EJB server) uses to alert that bean that some significant event in its lifecycle is about to, or just has, occurred. Of particular importance are the two notification methods (`ejbLoad( )` and `ejbStore( )`) used to alert the bean when its being synchronized with the database. The behavior of these methods changes depending on whether the bean is designed for BMP or CMP.

2. what is a session bean?

A session bean is a type of enterprise bean; a type of EJB server-side component. Session bean components implement the `javax.ejb.SessionBean` interface and can be stateless or stateful. Stateless session beans are components that perform transient services; stateful session beans are components that are dedicated to one client and act as a server-side extension of that client.
Session beans can act as agents modeling workflow or provide access to special transient business services. As an agent, a stateful session bean might represent a customer's session at an online shopping site. As a transitive service, a stateless session bean might provide access to validate and process credit card orders.
Session beans do not normally represent persistent business concepts like Employee or Order. This is the domain of a different component type called an entity bean.

3. what are the methods of Entity Bean?
4. How does Stateful Session bean store its state ?

When a client refers to a Stateful Session object reference, all calls are directed to the same object on the EJB container. The container does not require client identity information or any cookie object to use the correct object.
This means that for a client to ensure that calls are directed to the same object on the container, all it has to do is to use same reference for every call.
For example the following holds for all stateful session beans:

```
StatefulHome sfh = ...//get home interface for stateful bean
Stateful bean1 = sfh.create();
Stateful bean2 = sfh.create();
if (bean1.isIdentical(bean1)){} //this is true!
if (bean1.isIdentical(bean2)){} //this is false!

//Note that the second test would evaluate to true for stateless beans
```

Thus, if you're calling a Stateful Session Bean from a servlet, your servlet need to keep the reference to the remote object in the HttpSession object between client calls for you to be able to direct calls to the same object on the container.
Likewise, if you're calling from an application, you only obtain the reference to the bean once and reuse the object throughout the application session.

5. why does Stateless Session bean not store its state even though it has ejbActivate and ejbPassivate ?
   To have a consistent interface, so that there is no different interface that you need to implement for Stateful Session Bean and Stateless Session Bean.
   Both Stateless and Stateful Session Bean implement javax.ejb.SessionBean and this would not be possible if stateless session bean is to remove ejbActivate and ejbPassivate from the interface.
   You could argue that the two (stateful and stateless) are so different that they should have their own interface but Sun did not think so. They made both session beans implement the same interface and provided deployment descriptor to denote which one is it that you are deploying.

6. What are the services provided by the container ?
   An EJB container provides a full list of services (and technologies) like persistence,

transaction handling, security, communication between components and with other containers, caching, and so on.

Advantages of EJB
Its very simple question to answer but its a experience to understand it. I will try to answer : First of all, why session beans ? aren' t they simple java classes ? yes! they are ! But these are classes which are meant to be running in ejb server environment. whenever a http request comes , your servlet will create a objetc of the class and will use it and then it will be garbage collected. while in ejb server environment , u dont control the life of these objects rather server will create a pool before even first request comes and you are simply assigned one of these and taken back after u r finished. so instantiation and killing is avoided for each request.It saves a lot of processing and memory which is critical for any enterprise application. secondly, entity beans are answer to the persistance problem.You dont need to write embedded sql inside ur java classes to access database.It is handled by container software. thirdly, u dont need to haggle around with rmi problems , it is taken care of by ejb server. most important of all, it supports 2-phase commit. you can have a transaction spawned across multiple server connections and u dont need to code for this functionality. with ejb 2.0 coming up , OR mapping and message beans are introduced , so advantages are not countable.
Its oke that the server provides these services to EJB' s. But these advantages are more or less for the App server.What stops the App server from providing these services to simple java classes. Do they(EJB) have any feature that at runtime they become entitled for all these services or its up to the server to provide these services.
Actually , if u look at the ejb interfaces that u have to implement to create ejbs,
u will see that there are methods like ejbCreate(), ejbPostCreate(),setEntityContext() etc.,
These methods are life cycle methods which are actually being called by ejb
server to manage your ejb java object.
so ur ejb java classes are like any java class except that these
are following a protocol that helps server to talk to them
and let them use many services that ejb server provides,
simply by telling it either in xml format or by some java class that provides
this information in specific format.
these are called deployment discriptors.

7. Types of transaction ?
   there are total of 6 transactional attributes you can supply for an EJB method. transaction is managed through deployment descriptor. for each method in the entity bean you can specify transactional properties in the deployment descriptor. if all is done correctly, your code will not contain anything related to transactions in the entity beans themselves. you should not use UserTransaction in EJBs since the container will manage the transactions for EJBs for you based on your instructions in the deployment descriptor.
8.

What is bean managed transaction ?
The EJB specification says that we cannot use bean managed transaction in entity beans. I would like to know why we cant manage our transaction in entity beans.

The short, practical answer is ... because it makes your entity beans useless as a reusable component. Also, transaction management is best left to the application server - that's what they're there for.
It's all about atomic operations on your data. If an operation updates more than one entity then you want the whole thing to succeed or the whole thing to fail, nothing in between. If you put commits in the entity beans then it's very difficult to rollback if an error occurs at some point late in the operation.
Think of an account transfer which takes money from the first account and then pays it into a second account. If the paying in part fails how do you put the money back in the first account? What about if that then fails too?
The transaction should always "wrap" the entire operation. There are two obvious ways of achieving this in an EJB environment:
Use the javax.transaction package
Use a session bean to represent usecases and specify how transactions should be managed as part of the session bean's deployment descriptor.
Solution 2 is the "correct" way to do it. Either way, you can simply leave all transaction code out of your entity beans.

9.
10. Why does EJB needs two interface(Home and Remote Interface) ?
    the home interface, usually obtained via JNDI, is a factory mechanism used to create the remote. you use the remote interface to actually comminicate with the container.

In a few words, I would say that the main reason is because there is a clear division of roles and responsabilities between the two interfaces.
The *home* interface is your way to communicate with the container, that is who is responsable of creating, locating even removing one or more beans.
The *remote* interface is your link to the bean, that will allow you to remotely access to all its methods and members.
As you can see there are two distinct elements (the container and the beans) and you need two different interfaces for accessing to both of them.

11. What are transaction attributes ?
    Mr.A.J.P Transaction is communication between EJB and DB.This can be maintained in weblogic server by providing transaction attribute in xml file.
    Transaction is set of steps that should happen as an atomic step. i.e, suppose you have a bean that does the 1.reservation of ticket and 2.then charges the cerdit card.. now the two steps should either happen together or should not happen at all else you will end up with inconsistency in the system.. these two steps should be part of a transaction. this is handled by the container, in the ejb-jar.xml file, you mention the steps/methods that should be part of a transaction.

what is a container?
Answer

Enterprise beans are software components that run in a special environment called an EJB container. The container hosts and manages an enterprise bean in the same manner that a Java WebServer hosts a Servlet or an HTML browser hosts a Java applet. An enterprise bean cannot function outside of an EJB container. The EJB container manages every aspect of an enterprise bean at run time including remote access to the bean, security, persistence, transactions, concurrency, and access to and pooling of resources.

The container isolates the enterprise bean from direct access by client applications. When a client application invokes a remote method on an enterprise bean, the container first intercepts the invocation to ensure persistence, transactions, and security are applied properly to every operation a client performs on the bean. The container manages security, transactions, and persistence automatically for the bean, so the bean developer doesn't have to write this type of logic into the bean code itself. The enterprise bean can focus on encapsulating business rules, while the container takes care of everything else.

An enterprise bean depends on the container for everything it needs. If an enterprise bean needs to access a JDBC connection or another enterprise bean, it does so through the container; if an enterprise bean needs to access the identity of its caller, obtain a reference to itself, or access properties it does so through the container. The enterprise bean interacts with its container through one of three mechanisms: callback methods, the `EJBContext` interface, or JNDI.

Callback Methods: Every bean implements a subtype of the `EnterpriseBean` interface which defines several methods, called callback methods. Each callback method alerts the bean of a different event in its lifecycle and the container will invoke these methods to notify the bean when it's about to pool the bean, persist its state to the database, end a transaction, remove the bean from memory, etc. The callback methods give the bean a chance to do some housework immediately before or after some event. Callback methods are discussed in more detail in other sections.

EJBContext: Every bean obtains an `EJBContext` object, which is a reference directly to the container. The `EJBContext` interface provides methods for interacting with the container so that that bean can request information about its environment like the identity of its client, the status of a transaction, or to obtain remote references to itself.

JNDI: Java Naming and Directory Interface is a Java extension API for accessing naming systems like LDAP, NetWare, file systems, etc. Every bean automatically has access to a special naming system called the Environment Naming Context (ENC). The ENC is managed by the container and accessed by beans using JNDI. The JNDI ENC allows a bean to access resources like JDBC connections, other enterprise beans, and properties specific to that bean.

The EJB specification defines a bean-container contract, which includes the mechanisms (callbacks, `EJBContext`, JNDI ENC) described above as well as a strict set of rules that describe how enterprise beans and their containers will behave at runtime, how security access is checked, transactions are managed, persistence is applied, etc. The bean-container contract is designed to make enterprise beans portable between EJB containers so that enterprise beans can be developed once then run in any EJB container. Vendors like BEA, IBM, and Gemstone sell application servers that include EJB containers. Ideally, any enterprise bean that conforms to the specification should be able to run in any conformant EJB container.

Portability is central to the value that EJB brings to the table. Portability ensures that a bean developed for one container can be migrated to another if another brand offers more performance, features, or savings. Portability also means that the bean developer's skills can

be leveraged across several EJB container brands, providing organizations and developers with better opportunities.
In addition to portability, the simplicity of the EJB programming model makes EJB valuable. Because the container takes care of managing complex tasks like security, transactions, persistence, concurrency and resource management the bean developer is free to focus attention on business rules and a very simple programming model. A simple programming model means that beans can be developed faster without requiring a Ph.D. in distributed objects, transactions and other enterprise systems. EJB brings transaction processing and distributed objects development into the mainstream.

12. What methods do u use in Servlet – Applet communication ?
13. What are the types of Servlet ?
14. Difference between HttpServlet and Generic Servlets ?

    A servlet is a way of extending your web server with a Java program to perform tasks previously dealt with by CGI scripts or proprietary server extension frameworks. For more information, visit Sun's Servlet API home at
    GenericServlet is for servlets that might not use HTTP, like for instance FTP servlets. Of course, it turns out that there's no such thing as FTP servlets, but they were trying to plan for future growth when they designed the spec. Maybe some day there will be another subclass, but for now, always use HttpServlet.

Diff bet servlet and java beans?
JavaBeans are a set of rules to follow to create reusable software components, or beans. This contains properties and events. At the end you have a component which could be examined by a program (like an IDE) to allow the user of your JavaBean component to configure it and to run in its Java programs.
Servlets are Java classes running in a Servlet engine implementing a particular interface: Servlet, forcing you to implement some methods (service()). The servlet is an extension of your web server where this servlet is running on and only lets you know when a user requests a GET or POST calls from a web page to your servlet.
So, both have nothing in common except Java.

Diff bet serveltes and jsp?
Short answer: a JSP is a Servlet that thinks it's a Web page.
Medium answer: Both use server-side Java to dynamically generate web pages. The source code to a JSP looks like HTML, with Java embedded inside funny tags (*); the source code to a servlet looks like Java, with HTML embedded in out.print(...) statements. Both use the Servlet API to communicate with the web server and the client. In fact, a JSP gets compiled into a servlet, so they're almost identical in terms of expressive power. The choice is, whether you're more comfortable coding your pages in Java or in JSP-style HTML; and since you can call a JSP from a Servlet and vice versa, you don't have to make an either-or decision.
Long answer: See the Servlet FAQ and the JSP FAQ for all the information you need about both technologies.

```
(*) "Funny tags:" JSP can contain (a) normal HTML tags, (b) JSP tags
like <jsp:include>, (c) custom tags, (d) scriptlets (Java code
surrounded with <% and %>).
```

`What is a servlet engine?`
A "servlet engine" is a program that plugs in to a web server and runs servlets. The term is obsolete; the preferred term now is "servlet container" since that applies both to plug-in engines and to stand-alone web servers that support the Servlet API.

Difference between doGet and doPost ?
`doGet` is called in response to an HTTP GET request. This happens when users click on a link, or enter a URL into the browser's address bar. It also happens with some HTML FORMs (those with METHOD="GET" specified in the FORM tag).
`doPost` is called in response to an HTTP POST request. This happens with some HTML FORMs (those with METHOD="POST" specified in the FORM tag).
Both methods are called by the default (superclass) implementation of `service` in the `HttpServlet` base class. You should override one or both to perform your servlet's actions. You probably shouldn't override `service()`.

15.
16. What are the methods in HttpServlet?
17. What are the types of SessionTracking?
18. What is Cookie ? Why is Cookie used ?
    A cookie is a text-only string that gets entered into the memory of your browser. This value of a variable that a website sets. If the lifetime of this value is set to be longer than the time you spend at that site, then this string is saved to file for future reference.
    There are many reasons a given site would wish to use cookies. These range from the ability to personalize information (like on My Yahoo or Excite), or to help with on-line sales/services (like on Amazon Books or eBay), or simply for the purposes of collecting demographic information (like DoubleClick). Cookies also provide programmers with a quick and convenient means of keeping site content fresh and relevant to the user's interests. The newest servers use cookies to help with backend interaction as well, which can improve the utility of a site by being able to securely store any personal data that the user has shared with a site (to help with quick logins on your favorite sites, for example).
19. If my browser does not support Cookie,and my server sends a cookie instance What will happen ?
20. Why do u use Session Tracking in HttpServlet ?
21. Can u use javaScript in Servlets ?
22. What is the capacity the doGet can send to the server ?
23. What are the type of protocols used in HttpServlet ?
24. Difference between TCP/IP and IP protocol ?
25. Why do you use UniCastRemoteObject in RMI ?
26. How many interfaces are used in RMI?
27. Can Rmi registry be written in the code, without having to write it in the command prompt and if yes where?
28. Why is Socket used ?
29. What are the types of JDBC drivers ?
30. Explain the third driver(Native protocol driver) ?
31. Which among the four driver is pure Java driver ?
32. What are the Isolation level in JDBC transaction ?
33. How do you connect with the database ?
34. How do you connect without the Class.forName (" ") ?
35. What does Class.forName return ?
36. What are the types of statement ?
37. Why is preparedStatement,CallableStatement used for?

38. In real time project which driver did u use ?
39. Difference between AWT and Swing compenents ?
40. Is there any heavy weight component in Swings ?
41. Can the Swing application if you upload in net, be compatible with your browser?
42. What should you do get your browser compatible with swing components?
43. What are the methods in Applet ?
44. When is init(),start() called ?
45. When you navigate from one applet to another what are the methods called ?
46. What is the difference between Trusted and Untrusted Applet ?
47. What is Exception ?
48. What are the ways you can handle exception ?
49. When is try,catch block used ?
50. What is finally method in Exceptions ?
51. What are the types of access modifiers ?
52. What is protected and friendly ?
53. What are the other modifiers ?
54. Is synchronised modifier ?
55. What is meant by polymorphism ?
56. What is inheritance ?
57. What is method Overloading ? What is this in OOPS ?
58. What is method Overriding ? What is it in OOPS ?
59. Does java support multi dimensional arrays ?
60. Is multiple inheritance used in Java ?
61. How do you send a message to the browser in JavaScript ?
62. Does javascript support multidimensional arrays ?
63. Why is XML used mainly?
64. Do use coding for database connectivity in XML?
65. What is DTD ?
66. Is there any tool in java that can create reports ?

Differences between Mac, Windows, UNIX

MacOS (up until osX, anyhow) was single-user and (technically) did not have multitasking capabilites (some people will try to tell you that the mac has "cooperative multitasking", but I wouldn't even cosider that multitasking, since it requires each running program to voluntarily pause and wait for others to run).

Windows, starting with Win95 is a single-user, multasking OS. Although newer versions liek XP and 2000 support file permissions via EFS, the DO NOT allow multiple users to be logged in at once.

UNIX is 100% multi-user and multitasking, and is therefore inherently stable, secure, and confusing. Once you go UNIX, though, you'll never go back! UNIX (in all caps) was developed by AT&T waaaay back, Unix (no caps) usually refers to a family of operating systems (including the free-but-overrated Linux) that are based on UNIX. Solaris, BSD, OSF, SCO, and Tru64 are all flavors of Unix.

How does Windows programming differ from DOS programming?

Traditional dos programming is procedural, meaning that your program starts at one place, and steps through the code sequentially.

Windows programming is event-based, meaning that each time an event (like a button-click) occurs, a handler is invoked to produce a responce. The difference can be fustrating, since in event-driven code you can inadvertantly have several functions running simultaneously on the same data.

Windows - GUI DOS - CUI.
Windows - 32bit O.S and Dos is 16 bit O.S.
Windows - multithreading and multitasking O.S but Dos is stranger to these concepts.

What is the advantage of Windows?
GUI, Multitasking, Multithreaded O.S.

Windows' strongest advantge over Unix is its ability to feed system adminstrators.

Microsoft made sure that Windows was crash-prone enough to keep admins employed by service providers. In addition, M$ made sure that Windows' internals are kept top-secret and are closed-source so that sysadmins have to spend literally thousands of hours trying to network them to any existing non-windows platforms, thus ensuring that the sysadmin will have plenty of work to do for the forseeable future.

(The samba project is an excellent example....those guys spent thousands of hours reverse-engineering the proprietary SMB networking protocol to do something that would have taken only a few hours with NFS.)

What does winmain and wndproc do?

Define and explain COM
COM - Component Object Model.
It is a protocol for object communication from microsoft.

COM is a binary standard on Microsoft (and some UNIX platforms) that enables objects to interoperate in a networked environment regardless of the language in which they were developed or on which computers they reside. COM allows an object to expose its functionality to other components and to host applications. It defines both how the object exposes itself and how this exposure works across processes and across networks. COM also defines the object's life cycle.

Define Query Interface, Adref, and Release.

QueryInterface : This method in IUnknown lets you query a COM object if it supports a particular interface. If the object has inplemented the requested interface, it returns a valid interface pointer.

AddRef: Used for reference counting. Adds a reference count.

Release: Used for reference counting. Subtracts a reference count.

Do COM keep track of all the object references(Accounting)?
Yes, it positively does (using AddRef and Release). AddRef has to be called by the client, though.

What is Marshalling?
Marshalling is the process of converting data into a format for transfering over the network.

When is Marshalling not necessary?
Marshalling is used when Data is to be shared amoug different COM components. Not needed when they operate isolated.

Differences between Windows 3.1, Windows 95/98, and Windows NT
Windows 3.1 : 16-bit
Windows 9x : 32-bit
Windows NT : 32-bit, more secure, users must login

What is a thread?

A thread is the smallest unit of execution (having its own program counter).

Threads are light-weight processes. Thread switching takes lesser time than process switching.
Thread is a path of execution with a process.

How is a thread different from a process?

A process runs in its own address space. No two processes share their address space.

Threads will run in the same address space of the process that owns them. Threads belonging to the same process will run in the same address space. They will also share the process' s stack and heapNo two processes will share their stacks or their heaps.

Thread is a piece of execution i.e., a program

Process is a combination of Threads.

How is multithreading useful for a web server? A browser?

Web Servers are typically capable of serving multiple clients concurrently, this is achieved using Multi Threading

Maximum use of multiple treads will be made when there is more than one processor in the box.

However, on a single processor box the response time for each of the users may be better even if the overall throughput stays the same (will actually decrease slightly due to the processing time necessary to manage the threads)

Why not use multi-process for this? Why would you WANT to run a multi-process web server?

The reason you use a multi-process web server is for robustness in the face of failure. On a multithreaded server one thread can wipe out all the rest when it crashes the server. In a multi-process server only the offending process goes down. As far as speed is concerned most multi-process servers allow you to spawn at the begining any number of processes to handle requests so speed differences on a running system are not much different between multi-threading and multi-process servers.

Synchronize between threads vs. between processes?

Since threads are all part of a common process, they are much easier to sync. You could use a mutex or semaphore, but it is probably easier just to use global memory. All threads of a process can access global variables, so it can be quite easy to pass signals between them. POSIX threads support reenterant threads and joining, which always seemd like it defeated the purpose of having a thread in the first place.

Communicating between processes (or "IPC" for "Inter-Process Communication") is mostly OS-dependent. Under UNIX, pipes are probably the easiest way to do IPC. Sockets are somewhat portable, and allow communications to remote systems. There are countless other methods, of course.

sql

Q1. What is the basic difference between a **join** and a **union**?

A1. A **join** selects columns from 2 or more tables. A **union** selects rows.

Q2. What is **normalization** and what are the **five normal forms**?

A2. Normalization is a design procedure for representing data in tabular format. The five normal forms are progressive rules to represent the data with minimal redundancy.

Q3. What are **foreign keys**?

A3. These are attributes of one table that have matching values in a **primary key** in another table, allowing for relationships between tables.

Q4. Describe the elements of the SELECT query syntax.

A4. SELECT element FROM table WHERE conditional statement.

Q5. Explain the use of the WHERE clause.

A5. WHERE is used with a relational statement to isolate the object element or row.

Q6. What techniques are used to retrieve data from more than one table in a single SQL statement?

A6. **Joins, unions** and **nested selects** are used to retrieve data.

Q7. What is a **view**? Why use it?

A7. A view is a virtual table made up of data from base tables and other views, but not stored separately.

Q8. Explain an **outer join**.

A8. An outer join includes rows from tables when there are no matching values in the tables.

Q9. What is a **subselect**? Is it different from a **nested select**?

A9. A **subselect** is a select which works in conjunction with another select. A **nested select** is a kind of subselect where the inner select passes to the where criteria for the outer select.

Q10. What is the difference between **group by** and **order by**?

A10. **Group by** controls the presentation of the rows, **order by** controls the presentation of the columns for the results of the SELECT statement.

Q11. What keyword does an SQL SELECT statement use for a string search?

A11. The LIKE keyword allows for string searches. The % sign is used as a wildcard.

Q12. What are some sql **aggregates** and other **built-in functions**?

A12. The common **aggregate, built-in functions** are AVG, SUM, MIN, MAX, COUNT and DISTINCT.

Q13. How is the SUBSTR keyword used in sql?

A13. SUBSTR is used for string manipulation with column name, first position and string length used as arguments. Eg. SUBSTR (NAME, 1 3) refers to the first three characters in the column NAME.

Q14. Explain the EXPLAIN statement.

A14. The explain statement provides information about the optimizer's choice of access path of the sql.

Q15. What is **referential integrity**?

A15. **Referential integrity** refers to the consistency that must be maintained between primary and foreign keys, ie every **foreign key** value must have a corresponding **primary key** value.

Q16. What is a **NULL** value? What are the pros and cons of using NULLS?

A16. A **NULL** value takes up one byte of storage and indicates that a value is not present as opposed to a space or zero value. It's the DB2 equivalent of TBD on an organizational chartand often correctly portrays a business situation. Unfortunately, it requires extra coding for an application program to handle this situation.

Q17. What is a **synonym**? How is it used?

A17. A **synonym** is used to reference a table or view by another name. The other name can then be written in the application code pointing to test tables in the development stage and to production entities when the code is migrated. The synonym is linked to the AUTHID that created it.

Q18. What is an **alias** and how does it differ from a **synonym**?

A18. An **alias** is an alternative to a **synonym**, designed for a distributed environment to avoid having to use the location qualifier of a table or view. The alias is not dropped when the table is dropped.

Q19. When can an insert of a new primary key value threaten **referential integrity**?

A19. Never. New primary key values are not a problem. However, the values of foreign key inserts must have corresponding primary key values in their related tables. And updates of primary key values may require changes in foreign key values to maintain **referential integrity**.

Q20. What is the difference between **static** and **dynamic sql**?

A20. Static sql is hard-coded in a program when the programmer knows the statements to be executed. For **dynamic sql** the program must dynamically allocate memory to receive the query results.

Q21. Compare a **subselect** to a **join.**

A21. Any subselect can be rewritten as a join, but not vice versa. Joins are usually more efficient as join rows can be returned immediately, subselects require a temporary work area for inner selects results while processing the outer select.

Q22. What is the difference between **IN subselects** and **EXISTS** subselect?

A22. If there is an index on the attributes tested an IN is more efficient since DB2 uses the index for the IN. (IN for index is the mnemonic).

Q23. What is a **Cartesian product**?

A23. A **Cartesian product** results from a faulty query. It is a row in the results for every combination in the join tables.

Q24. What is a **tuple**?

A24. A **tuple** is an instance of data within a relational database.

Q25. What is the difference between **static** and **dynamic sql**?

A25. Static sql is compiled and optimized prior to its execution; **dynamic** is compiled and optimized during execution.

Q26. Any SQL implementation covers data types in couple of main categories. Which of the following are those data types ? (Check all that apply) A. NUMERIC B. CHARACTER C. DATE AND TIME D. BLOBS E. BIT

A26. A,B,C. Not all SQL implementations have a BLOB or a BIT data types.

Q.27. We have a table with a CHARACTER data type field. We apply a ">" row comparison between this field and another CHARACTER field in another table. What will be the results for records with field value of NULL ? (Check one that applies the best) A. TRUE B. FALSE C. UNKNOWN D. Error. E. Those records will be ignored

A27. C. NULL in a row when compared will give an UNKNOWN result.

Q.28. Any database needs to go through a **normalization** process to make sure that data is represented only once. This will eliminate problems with creating or destroying data in the database. The normalization process is done usually in three steps which results in first, second and third normal forms. Which best describes the process to obtain the third normal form? (Check one that applies the best) A. Each table should have related columns. B. Each separate table should have a primary key. C. We have a table with multi-valued key. All columns that are dependent on only one or on some of the keys should be moved in a different table. D. If a table has columns not dependent on the primary keys, they need to be moved in a separate table. E. Primary key is always UNIQUE and NOT NULL.

A28. D. All columns in a table should be dependent on the **primary key.** This will eliminate transitive dependencies in which A depends on B, and B depends on C, but we' re not sure how C depends on A.

Q.29. SQL can be embedded in a host program that uses a relational database as a persistent data repository. Some of the most important pre-defined structures for this mechanism are SQLDA ("SQL Descriptor Area") and SQLCA ("SQL Communications Area") SQLCA contains two structures - SQLCODE and SQLSTATE. SQLSTATE is a standard set of error messages and warnings in which the first two characters defines the class and the last three defines the subclass of the error. Which of the following SQLSTATE codes is interpreted as "No data returned"? (Check one that applies the best) A. 00xxx B. 01xxx C. 02xxx D. 22xxx E. 2Axxx

A29. C. 00 - is successful completion, 01 - warnings, 22 - is data exception and 2A is syntax error. The SQLSTATE code format returned for "No data returned" is "02xxx".

Why is a manhole cover round?
There are quite a few possible answers to this question. Some are:
1. The Man hole cover being roundAnswer
The Man hole cover is round so it will not fall into the hole. If it were square it would be possible for the cover to fall into the hole if it was tilted at the proper angle. Being round there is no way without breaking the cover for it to fall into the hole (sent by Tony Bowman)
2. Because manholes are round. (Simple and seemingly obvious, yet absolutely true)
3. Round covers can be transported by one person, because they can be rolled on their edge. (Ease of use).
4. A round cover doesn't need to be rotated to fit over a hole

Given a rectangular cake with a rectangular piece removed (any size or orientation), how would you cut the remainder of the cake into two equal halves with one straight cut of a knife ?

Find the centers of both the original cake and the removed piece. Cut the reminder
along the line connecting these two centers.

This is true because this line cut both the
original cake and the removed piece in half,
thus the remainder into two halves.

: If you had an infinite supply of water and a 5 quart and 3 quart pail, how would you measure exactly 4 quarts?

Measure 5 quart using the 5 quart pail, Then fill the 3 quart pail from 5 quat p
ail. You have 2 quart left in the 5 quart pail. So, measure 2 quarts. Repeat the
process and you will have measured four quarts.

One train leaves Los Angeles at 15 MPH heading for New York. Another train leaves from New York at 20mph heading for Los Angeles on the same track. If a bird, flying at 25mph, leaves from Los Angeles at the same time as the train and flies back and forth between the two trains until they collide, how far will the bird have traveled?

1. Let's say the distance between LA and NY is d miles.

2. The time before which the trians will collide : d / (15+20) hours.

3. Distance travelled by the bird in that time : (d / 35) * 25 = 5*d/7 miles.

There are four people who need to cross a bridge at night. The bridge is only wide enough for two people to cross at once. There is only one flashlight for the entire group. When two people cross, they must cross at the slower member's speed. All four people must cross the bridge in 17 minutes, since the bridge will collapse in exactly that amount of time. Here are the times each member takes to cross the bridge:

- Person A: 1 minute
- Person B: 2 minutes
- Person C: 5 minutes
- Person D: 10 minutes

So, if Person A and C crossed the bridge initally, 5 minutes would elapse, because Person C takes 5 minutes to cross. Then, Person A would have to come back to the other side of the bridge, taking another minue, or six minutes total. Now, if Person A and D crossed the bridge next, it would take them 10 minutes, totalling to 16 minutes. If A came back, it would take yet another minute, totally 17... the bridge would collapse with A and B on the wrong side. How can all four people get across the bridge within 17 minutes? *Note: there is no trick-answer to this problem. You can't do tricky stuff like throuing the flashlight back from one end of the bridge to the other. This problem can be solved!*

Yeah, this problem is little bit more tricky...but that's wht's makes it interesting.. :)

1. A and B cross together. Total Time: 2 Minutes
2. A comes back. Total Time: 3 Minutes.
3. C and D cross together. Total Time: 13 Minutes
4. B comes back. Total Time: 15 Minutes
5. A and B cross together. Total Time: 17 Minutes

You have someone working for you for seven days and a gold bar to pay them. The gold bar is segmented into seven connected pieces. You must give them a piece of gold at the end of every day. If you are only allowed to make two breaks in the gold bar, how do you pay your worker?

Day 1 : Break of a single segment and give it to the worker.
Day 2 : Break two segments from the bar, Ask the worker to return the segment you gave him on day 1 and give him the segment with Two connected pieces.
Day 3 : GIve the worker the 1 piece segment you have.
Day 4 : Ask the worker to return all three pieces he has and give him the segment with 4 pieces on it.
Day 5 : Give him the the segment with 1 piece.
Day 6 : Ask him to return the 1 piece segment and give him the segment with two pieces.
Day 7 : Give him the 1 piece segment you have.

Important point is to notice this pattern here :
1
10
11
100
101
110
111

Now, you can answer this question for any bar with "n" segments.

You have 5 jars of pills. Each pill weighs 10 gram, except for contaminated pills contained in one jar, where each pill weighs 9 gm. Given a scale, how could you tell which jar had the contaminated pills in just one measurement?
1. Mark the jars with numbers 1, 2, 3, 4, and 5.
2. Take 1 pill from jar 1, take 2 pills from jar 2, take 3 pills from jar 3, take 4 pills from jar 4 and take 5 pills from jar 5.
3. Put all of them on the scale at once and take the measurement.
4. Now, subtract the measurment from 150 (1*10 + 2*10 + 3*10 + 4*10 + 5*10)
5. The result will give you the jar number which has contaminated pill.

You have 12 balls. All of them are identical except one, which is either heavier or lighter than the rest - it is either hollow while the rest are solid, or solid while the rest are hollow. You have a simple two-armed scale, and are permitted three weighings. Can you identify the odd ball, and determine whether it is hollow or solid? (This question is same as above but with an additional twist).

I had already worked this problem out for someone else using pills.
The setting of the problem, as I remember it, goes something like
this.

A wise man has committed a capital crime and is to be put to death.
The king decides to see how wise the man is. He gives to the man 12
pills which are identical in size, shape, color, smell, etc. However,
they are not all exactly the same. One of them has a different
weight. The wise man is presented with a balance and informed that all
the pills are deadly poison except for the one which has a different
weight. The wise man can make exactly three weighings and then must
take the pill of his choice.

Remember that the only safe pill has a different weight, but the
wise man doesn't know whether it is heavier or lighter.

Most people seem to think that the thing to do is weigh six pills against six pills, but if you think about it, this would yield you no information concerning the whereabouts of the only safe pill.

So that the following plan can be followed, let us number the pills from 1 to 12. For the first weighing let us put on the left pan pills 1,2,3,4 and on the right pan pills 5,6,7,8.

There are two possibilities. Either they balance, or they don't. If they balance, then the good pill is in the group 9,10,11,12. So for our second weighing we would put 1,2 in the left pan and 9,10 on the right. If these balance then the good pill is either 11 or 12.

Weigh pill 1 against 11. If they balance, the good pill is number 12. If they do not balance, then 11 is the good pill.

If 1,2 vs 9,10 do not balance, then the good pill is either 9 or 10. Again, weigh 1 against 9. If they balance, the good pill is number 10, otherwise it is number 9.

That was the easy part.

What if the first weighing 1,2,3,4 vs 5,6,7,8 does not balance? Then any one of these pills could be the safe pill. Now, in order to proceed, we must keep track of which side is heavy for each of the following weighings.

Suppose that 5,6,7,8 is the heavy side. We now weigh 1,5,6 against 2,7,8. If they balance, then the good pill is either 3 or 4. Weigh 4 against 9, a known bad pill. If they balance then the good pill is 3, otherwise it is 4.

Now, if 1,5,6 vs 2,7,8 does not balance, and 2,7,8 is the heavy side, then either 7 or 8 is a good, heavy pill, or 1 is a good, light pill.

For the third weighing, weigh 7 against 8. Whichever side is heavy is the good pill. If they balance, then 1 is the good pill. Should the weighing of 1,5, 6 vs 2,7,8 show 1,5,6 to be the heavy side, then either 5 or 6 is a good heavy pill or 2 is a light good pill. Weigh 5 against 6. The heavier one is the good pill. If they balance, then 2 is a good light pill.

I think that one could write all of this out in a nice flow chart, but I'm not sure that the flow chart would show up correctly over e-mail.

After trying to solve the problem without cheating, I found a solution. It covers the case Mr. Manning does not cover and is hopefully easier to follow than Dr. Robert's solution.

If you have balls numbered 1-12 and if 1-4 and 5-8 don't balance (otherwise see above) then remove 1-3 from the left side. Replace with 5-7 (now containing 4-7). Add 9-11 to the right side (now containing 8-11).

If the scales balance then 1-3 contains the odd ball (heavier or lighter depending on whether the left side was heavier or lighter).

If the scales reverse then 5-7 contain the odd ball, which is deduced to be heavier or lighter as above.

In either case you have 3 balls and since you know if you are looking for a heavy or a light ball you can determine which ball it is in one weighing.

If the scales are unchanged then one of the unmoved balls (4 or 8) are odd. Weigh the heavier (say 4) against a known ball (say 12). If they balance, 8 is the odd (in this case lighter) ball. If they don't balance then the heavier ball is the odd ball.

If you are on a boat and you throw out a suitcase, will the level of water increase?

The answer is: it depends on the density of the briefcase.

While the briefcase is in the boat, it will displace it's weight in water.

If the briefcase floats after you toss it out, then it still displaces its weight in water, and there is no change.

However, if the briefcase is heavier than water it will displace its volume, not its weight. The amount of displaced water will be less (which is why it sank) and the water level will decrease.

There are 3 ants at 3 corners of a triangle, they randomly start moving towards another corner. What is the probability that they don't collide?

1. Let's mark the corners of the triangle as A,B,C.
   2. There are only two cases under which the ants won't collide : A->B, B->C, C->A and A->C, C->B and B->A
   3. Therefore, probability of ants not colliding : 2/8 = ¼

This puzzle was apparently written by Einstein in the last century. He said that 98% of the people in the world cannot solve the quiz. See if you can. (Sent by crit crit).

Facts:

1: There are 5 houses in 5 different colors
2: In each house lives a person with a different nationality.
3: These 5 owners drink a certain beverage, smoke a certain brand of cigar and keep a certain pet.
4: No owner has the same pet, smoke the same brand of cigar or drink the same drink.

HINTS:

1: The British lives in a red house.
2: The Swede keeps dogs as pets
3: The Dane drinks tea
4: The green house is on the left of the white house (it also means they are next door to each other)
5: The green house owner drinks coffee
6: The person who smokes Pall Mall rears birds
7: The owner of the yellow house smokes Dunhill
8: The man living in the house right in the center drinks milk
9: The Norwegian lives in the first house
10: The man who smokes Blend lives next to the one who keeps cats
11: The man who keeps horses lives next to the man who smokes Dunhill
12: The owner who smokes Blue Master drinks beer
13: The German smokes Prince
14: The Norwegian lives next to the blue house
15: The man who smokes Blend has a neighbor who drinks water.

The question is: WHO KEEPS THE FISH?

ONLY for those who want to know the approach for solving such problems

Ok first thing to figure out is the order of houses

we can use to fact
1) Green and White are next to each other
2) blue is the second house.

center house is imporatnt becase we have a clue about it

So the possiblilites could be:-

____, blue, ____, Green, White,

or
____, blue, Green, White,____

there are 2 blanks and two colors.

Red can't be the first house because, firts house should have Norway guy andRed house has British.

So we have :-

Yellow, blue, ____, Green, White,

or
Yellow, blue, Green, White,____
(where blank is Red)

2nd option can't be true as we have cluse that center house owner drinks milk, but the green house owner drinks coffee.

So we get the order as follow

Yellow, blue,Red, Green, White,
(with last two houses could be switched with no effect)

Then make the table as follows

house/attributes|Yellow|Blue|Red|Green|White|
Drink
Pet
Smoke
Nationality

Once you have the right order then its only the matter of reading the clues and filling in the table entries.

There are three wise men in a room. So you decide to give them a challenge. Suspecting that the thing they care about most is money, you give them $100 and tell them they are to divide this money observing the following rule: they are to discuss offers and counter-offers from each other and then take a vote. Majority vote wins. Sounds easy enough... now the question is, assuming each person is motivated to take the largest amount possible, what will the outcome be? (submitted by : mikeprimeau@ottawa.com)
2 probably won't vote the 3rd out, since counteroffers are allowed. Consider: A and B decide to leave C out and split halfway. C offers B to leave A out instead, and as incentive offers B $60 as compared to $50 (Note, for C, $40 is better than nothing at all). Now A will try to raise that figure, but keep no less than $33 for himself. It would also happen that as B starts to get more and more of the share, A and C would decide to keep B out instead and split amongst each other. This would go on and on until they reach equilibrium with each agreeing to take their rightful share of 33 dollars.

Three men were renting a motel figuring the room cost 30 dollars they would pitch in ten a piece.The room was only 25 so they each gave the bell boy ten,(tip)the bellboy didn''t think that would be fair so he gave them each back 1 dollar and kept 2 for himself.What happened to the other dollar? (sent by MACsabastion@webtv.net)

Oh after giving it some thought actually I was right, there isn't any missing dollar.

If you go backwards you come up to this:
9x3 = 27
27+2 = 29
so it had to be 30$ in order to be correct right?
No, because you have to add 3$ and not 2. You think it would be right if you add 2$ because the tip is that amount. But the tip is already included in the multiplication.
So the maths aren't right, it should be 9x3+3 and not 9x3+2.

"You have b boxes and n dollars. If I want any amount of money from 0 to n dollars, you must be able to hand me 0 to b boxes so that I get exactly what I request." The two questions were "What are the restrictions on b and n, and how is money distributed among the boxes?"

No one mentioned why one should use binary numbers !!
Any numner n can be expressed in any base.
n = a1 * b^0 + a2 * b^1 + a3 * b^2 + ...
where ai's are from 0 to b1.
eg. in decimal system
123 = 10^2 * 1 + 10 * 2 + 3.

In this case you either choose a box or not. So the ai's can be 0 or1.(0 means dont choose and 1 means you chose it)
Hence b = 2;
So binary base has to be used.

max that you can represent using this is
2^b - 1 >= n;
It is the relation between b and n.

The trick is of Binary Numbers
2^0= 1
2^1=2
2^2=4
2^3=8
2^4= 16
....
...

2^7= 128
...

..

The formula is [log n to base 2] + 1 = b
where [] denote integral part of the log base 2 of n

Example
So if we want 0 to 15 dollars
[log of 15 to base 2] + 1 = [3.90] + 1 = 3 + 1 = 4 boxes
we need to have 4 boxes each having 1,2,4,8, dollars respectively.Now we know from binary numbers that any amount from 0 to 15 can be formed with this boxes.

For 0 to 6 dollars
[log of 7 to base 2] + 1 = [2.58] + 1 = 2 + 1 = 3 boxes
we need to have 3 boxes each having 1,2,4, dollars respectively

For 0 to 100 dollars
[log of 100 to base 2] + 1 = [6.64] + 1 = 6 + 1 = 7 boxes
we need to have 7 boxes each having 1,2,4,8,16,32,64 dollars respectively

There are 2 rooms. First has 3 switches (on/off) and the other one has 3 bulbs (electric lamps). Initially all the switches are off and all bulbs are turned off. Every bulb is connected to one switch and only one. First you are in the room with switches. You can play with switches however you want and whenever you want, but in the rooms with bulbs you can enter only once. Find the correspondence between switches and bulbs. - (Sent by Oliviu Burlacu)

Just turn one light switch on(switch1) and leave it on long enough for the bulb to get warm. Then turn that off and turn another one on(switch2). Then go downstairs and the light that is currently on belongs to (switch2) and then a bulb that is off will be warm since you left it on for awhile so that belongs to (switch1) then the third bulb that is off belongs to the last switch that wasn't touched(switch3)

There is a prisioner who has two doors infront of him one leads to freedom and one leads to hell, he has no idea which one is which, but there is 2 guards there on tell the truth and one always lies. You can ask them one question that will lead you to your freedom. What is your question? - (Sent by Karl Hamaoui)

If you are allowed to ask one question per guard then ask them individually "what will the other guard say if I were to ask him the right door ?"

When this questioned is posed to the guard who speaks the truth, he will know that the other guard lies so will claim his gurad to be the right door. so the gurad who speaks the truth will point to the gurad who lies.

When this question is posed to the bad guard then without hesitation he would say that his door his right. From the above answers one can determine the right door.

This answer assumes that One question can be asked to both the guards, if not we need to call both the gurads and pose this question at the same time.

This is NOT my answer but found it as a riddle in a greek story (if my memory holds good)

Also this answer assumes that the guard who speaks the truth knows that the other guard lies and vice versa

If you look at a clock and the time is 3:15, what is the angle between the hour and the minute hands? (The answer to this is not zero!)

Obviously, for digital clock, nothing is meaningful in this problem.

For analog clock, the minute pin is exact at "3" and hour pin is a little past "3". Then tha angle is:
15/60 * (360/12)= 7.5 (degree)

where 15/60 = 15 minutes count in an hour(the angle between "3" and "4");
360/12 = a circle devided by 12 digits.

Notice that we have to find the angle swept by the hour hand when it was at 3 o' clock and when it is at 3:15.

This is so because the minute hand points to the same location at 3:15 as the hour hand was pointing at 3 o' clock.

In other words we have to find the angle swept by the hour hand in 15 minutes.

We know hour hand sweeps 360 degrees in 12 hours(12*60 minutes).

so, number of degrees it sweeps in 15 minutes = 15 * (360 / (12*60)) = 7.5 degrees.

i am posting this solution as i feel this is a simpler way to approach the problem.

bye,

What is the sum of the numbers from 1 to 1000?
In general the sum of numbers from 1 to n is given by: n*(n+1)/2

An easier way to remember it might be "times a half, plus a half", which is essentially what that calculation reduces to:

(n(n+1))/2 = n*(n/2) + n/2

You are an employer. You have ten employees. Each month, each one of your ten employees gives you ten bags of gold. Each bag of gold has ten pieces of gold in it. Each piece of gold weighs one pound. One of your employees is cheating you by only putting nine pieces of gold in each of his ten bags of gold. You have a scale (not a balance, a scale), and you can only take one measurement from the scale, only one (1) reading.

How can you tell which of the ten employees is cheating you by using this scale and only taking one measurement?

When your employees come to pay you at the end of the month, take one bag from employee number one's pile and set it aside. Take two bags from employee number two's pile and add it to the one from employee number one. Take three from employee three, four from employee four, etc...

Once you have now compiled a pile of bags, you should have 55 bags (one from one, two from two, etc). Set the whole pile (all 55 bags) on the scale and take a reading. If all the bags were honest, you would have 550 pounds on the scale.

If employee one is cheating you, then the reading would be 549, since you only have one bag from employee one. Likewise, if the measurement is 547, then 3 pounds are missing, and employee three is the thief.

How many points are there on the globe where by walking one mile south, one mile east and one mile north you reach the place where you started.

Imagine that you have 26 constants, labelled A through Z. Each constant is assigned a value in the following way: A = 1; the rest of the values equal their position in the alphabet (B corresponds to the second position so it equals 2, C = 3, etc.) raised to the power of the preceeding constant value. So, B = 2^(A's value), or B = 2^1 = 2. C = 3^2 = 9. D = 4^9, etc., etc. Find the exact numerical value to the following equation:
(X - A) * (X - B) * (X - C) * ... * (X - Y) * (X - Z)

(X - A) * (X - B) * (X - C) * ... * (X - Y) * (X - Z) equals 0 since (x-x) is zero.

(X - A) * (X - B) * (X - C) * ... * (X - Y) * (X - Z) = 0 as there is (X-X) which is equal to 0 and anything multiplied by 0 is 0.

You have two jars, 50 red marbles and 50 blue marbles. A jar will be picked at random, and then a marble will be picked from the jar. Placing all of the marbles in the jars, how can you maximize the chances of a red marble being picked? What are the exact odds of getting a red marble using your scheme?

zeephro is very close to being right.
sprihak's math is off.

Put 1 red marble in the first jar.
Put the other 49 reds and the 50 blues in the other jar (99 total marbles!).

The probability of selecting the jar with only 1 red marble and getting the only marble in the jar is 50% (1/2 * 1) not 1%!

The probability of selecting the other jar is 50% and the odds of getting a red marble from that jar is 49/99 and (1/2 * 49/99) = 24.7475%.

The total probability is 74.7475%. Zeephro forgot to correct for the missing red marble in the denominator (49/99, not 49/100) which is why he came out with 74.5%

Short answer: In one of the jars put 1 red ball and 0 blue balls, in the second jar put the rest of the balls.
The probabily of picking a red ball is 0.7474...

Long answer:
Let R1 and B1 be the number of Red and Blue balls in the first jar, and R2 and B2 the number of Red and Blue balls in the second jar.
Then a) R1 + R2 = 50 and b) B1 + B2 = 50.
The probabily of picking a Red ball is equal to the probabily of choosing a jar times the probabity of choosing a Red ball in that jar.

ProbRed = (prob of jar1) (prob of red in jar1) + (prob of Jar2)(prob of red in jar2)

The probability of each jar is 0.5 so:

c) ProbRed = 0.5 (Prob of red in jar1 + Prob of red in Jar2)

ProbRed1 = (num of red balls in jar1) /(total number of balls in jar1)
ProbRed1 = R1/(R1+B1) , the same for jar 2:
ProbRed2 = R2/(R2+B2)

If we replace a), b) in c) we get:

ProbRed = 0.5((R1/(R1+B1)) + ((50-R1) /(50-R1+50-R1)))

Now if you are good at math you can find the derivate of this function and equal it to zero, but I am not that good at math so I run a VB program for all posible values of R1 and B1 (except those that cause division by zero):

```
Private Sub Command1_Click()
Dim R1 As Integer
Dim B1 As Integer
Dim ProbRed As Double

For R1 = 1 To 49
For B1 = 0 To 50
ProbRed = 0.5 * ((R1 / (R1 + B1)) + ((50 - R1) / (50 - R1 + 50 - B1)))
List1.AddItem "R1 = " & R1 & " B1 = " & B1 & " ProbRed = " & ProbRed
Next B1
Next R1
End Sub
```

The result confirmed that the maximum probabity is when :
a) R1 = 1 and B1 = 0 or when
b) R1 = 49 and B1 = 50

Hope you found this semi-empiric demonstration useful. When I interview candidates I ask them to solve the problem or write a program to solve it...

Imagine you are standing in front of a mirror, facing it. Raise your left hand. Raise your right hand. Look at your reflection. When you raise your left hand your reflection raises what appears to be his right hand. But when you tilt your head up, your reflection does too, and does not appear to tilt his/her head down. Why is it that the mirror appears to reverse left and right, but not up and down?

It may be because, our eyes are placed left to right when facing the mirror. So the reflection from mirror crosses like a X in front of us in horizontal direction. As far as vertical reflection goes, the eyes are in same line. So the vertical reflection will be just like a horizontal line without reversals. That's why the image appears reversed horizontally and not vertically.

You have a bucket of jelly beans. Some are red, some are blue, and some green. With your eyes closed, pick out 2 of a like color. How many do you have to grab to be sure you have 2 of the same?

It is quite simple. If you only take 4 jelly beans, the possible combos that you can come up with are, in terms or a,b and c:

a,b,c,a a,b,c,b a,b,c,c
a,a,b,c a,a,b,b a,a,c,c
b,b,a,c b,b,a,a b,b,c,c
c,c,a,c c,c,a,a c,c,b,b
a,a,a,b a,a,a,c a,a,a,a
b,b,b,a b,b,b,c b,b,b,b
c,c,c,a c,c,c,b c,c,c,c

You are given a scale which you are to use to measure eight balls. Seven of these balls have the same weight: the eigth ball is heavier than the rest. What is the minimum number of weighs you could perform to find the heaviest of the eight balls?. Remmber it's a scale not a balance. (i.e. It can just tell you if one side is heavier than the other it can't give you the exact weight).

Good try!..Umm..You can do this in 2 weighs. Put three balls on each side of the scale. That's a total of six balls you're weighing. If the three balls on each side weigh equally, you know that one of the two remaining balls is the heaviest. Weigh those two balls to determine which one is heaviest. If, however, one of the three ball combinations weighs most, remove all balls from the scale, then weigh just two of the three "heavier" balls. If those two balls are equal weight, the third, unweighed ball is the heaviest; otherwise the scale will indicate which of the two balls on the scale is the heavy one.

There was a sheriff in a town who caught three outlaws. He said he was going to give them all a chance to go free. All they had to do is figure out what color hat they were wearing. The sheriff had 5 hats 3 black and 2 white. The first guy guessed and was wrong so he was put in jail. The second guy also guessed and was also put in jail. finally, a third blind man guessed and he guessed correctly. How did he know?

Humm, I puzzled over this for a sec. and then, when checking my answer, I had a hard time understanding what was written (though it is indeed correct) so here is a more lengthy answer that hopefully explains it all.

lets look at it this way; Here are our possibilities:
1 BBB
2 BBW
3 BWB
4 WBB
5 WWB
6 BWW
7 WBW

Ok, now we can eliminate # 6 because in this case convict one would be sure to know that he had on a white hat.

7 can be eliminated for the same reason for convict 2's guess.

In case # 2, the first convict has to see at least 1 black hat (if he saw two whie hats he wouldn't have guessed wrong guessing he had a white hat since he would already have seen 2 white hat). So we know that convict 2 or convict 3 has a black hat (posibaly both). Now convict two has the same delema, but he knows that one or both of convict 2 and 3 has a black hat. He can see that convict 3 has a white hat so in that case he would guess black and be correct, but he didn't (since we know he guessed wrong). given this, we can remove option 2 from consideration.

options 1,3,4,5 all have convict 3 wearing a black hat. Thus, assuming that convicts 1 and 2 are as logical as possible, the only options left all have convict 3 wearing a black hat.

What is SQL and where does it come from?

Structured Query Language (SQL) is a language that provides an interface to relational database systems. SQL was developed by IBM in the 1970s for use in System R, and is a de facto standard, as well as an ISO and ANSI standard. SQL is often pronounced SEQUEL.

In common usage SQL also encompasses DML (Data Manipulation Language), for INSERTs, UPDATEs, DELETEs and DDL (Data Definition Language), used for creating and modifying tables and other database structures.

The development of SQL is governed by standards. A major revision to the SQL standard was completed in 1992, called SQL2. SQL3 support object extensions and will be (partially?) implemented in Oracle8.

- Back to top of file

What are the difference between DDL, DML and DCL commands?

DDL is Data Definition Language statements. Some examples:

- CREATE - to create objects in the database
- ALTER - alters the structure of the database
- DROP - delete objects from the database
- TRUNCATE - remove all records from a table, including all spaces allocated for the records are removed
- COMMENT - add comments to the data dictionary
- GRANT - gives user's access privileges to database
- REVOKE - withdraw access privileges given with the GRANT command

DML is Data Manipulation Language statements. Some examples:

- SELECT - retrieve data from the a database
- INSERT - insert data into a table
- UPDATE - updates existing data within a table
- DELETE - deletes all records from a table, the space for the records remain
- CALL - call a PL/SQL or Java subprogram
- EXPLAIN PLAN - explain access path to data
- LOCK TABLE - control concurrency

DCL is Data Control Language statements. Some examples:

- COMMIT - save work done
- SAVEPOINT - identify a point in a transaction to which you can later roll back
- ROLLBACK - restore database to original since the last COMMIT
- SET TRANSACTION - Change transaction options like what rollback segment to use

- Back to top of file

How can I eliminate duplicates values in a table?

Choose one of the following queries to identify or remove duplicate rows from a table leaving one record:

Method 1:

```
SQL> DELETE FROM table_name A WHERE ROWID > (
  2    SELECT min(rowid) FROM table_name B
  3    WHERE A.key_values = B.key_values);
```

Method 2:

```
SQL> create table table_name2 as select distinct * from table_name1;
SQL> drop table_name1;
SQL> rename table_name2 to table_name1;
```

Method 3: (thanks to Kenneth R Vanluvanee)

```
SQL> Delete from my_table where rowid not in(
SQL>     select max(rowid) from my_table
SQL>     group by my_column_name );
```

Method 4: (thanks to Dennis Gurnick)

```
SQL> delete from my_table t1
SQL> where  exists (select 'x' from my_table t2
SQL>                where t2.key_value1 = t1.key_value1
SQL>                  and t2.key_value2 = t1.key_value2
SQL>                  and t2.rowid      > t1.rowid);
```

Note: If you create an index on the joined fields in the inner loop, you, for all intents purposes, eliminate N^2 operations (no need to loop through the entire table on each pass by a record).

- Back to top of file

How can I generate primary key values for my table?

Create your table with a NOT NULL column (say SEQNO). This column can now be populated with unique values:

```
SQL> UPDATE table_name SET seqno = ROWNUM;
```

or use a sequences generator:

```
SQL> CREATE SEQUENCE sequence_name START WITH 1 INCREMENT BY 1;
SQL> UPDATE table_name SET seqno = sequence_name.NEXTVAL;
```

Finally, create a unique index on this column.

- Back to top of file

How can I get the time difference between two date columns

Look at this example query:

```
select floor(((date1-date2)*24*60*60)/3600)
       || ' HOURS ' ||
       floor((((date1-date2)*24*60*60) -
       floor(((date1-date2)*24*60*60)/3600)*3600)/60)
       || ' MINUTES ' ||
       round((((date1-date2)*24*60*60) -
       floor(((date1-date2)*24*60*60)/3600)*3600 -
       (floor((((date1-date2)*24*60*60) -
       floor(((date1-date2)*24*60*60)/3600)*3600)/60)*60)))
       || ' SECS ' time_difference
from   ...
```

- Back to top of file

How does one count different data values in a column?

```
        select dept, sum(  decode(sex,'M',1,0)) MALE,
                     sum(  decode(sex,'F',1,0)) FEMALE,
                     count(decode(sex,'M',1,'F',1)) TOTAL
        from   my_emp_table
        group  by dept;
```

- Back to top of file

How does one count/sum RANGES of data values in a column?

A value x will be between values y and z if GREATEST(x, y) = LEAST(x, z). Look at this example:

```
        select f2,
               sum(decode(greatest(f1,59), least(f1,100), 1, 0)) "Range
60-100",
               sum(decode(greatest(f1,30), least(f1, 59), 1, 0)) "Range
30-59",
               sum(decode(greatest(f1, 0), least(f1, 29), 1, 0)) "Range
00-29"
        from   my_table
        group  by f2;
```

For equal size ranges it might be easier to calculate it with DECODE(TRUNC(value/range), 0, rate_0, 1, rate_1, ...). Eg.

```
select ename "Name", sal "Salary",
       decode( trunc(f2/1000, 0), 0, 0.0,
                                  1, 0.1,
                                  2, 0.2,
                                  3, 0.31) "Tax rate"
from   my_table;
```

- Back to top of file

Can one retrieve only the N th row from a table?

Ravi Pachalla provided this solution to select the Nth row from a table:

```
SELECT f1 FROM t1
WHERE  rowid = (
   SELECT rowid FROM t1
   WHERE  rownum <= 10
  MINUS
   SELECT rowid FROM t1
   WHERE  rownum < 10);
```

Alternatively...

```
SELECT * FROM emp WHERE rownum=1 AND rowid NOT IN
   (SELECT rowid FROM emp WHERE rownum < 10);
```

Please note, there is no explicit row order in a relational database. However, this query is quite fun and may even help in the odd situation.

- Back to top of file

Can one retrieve only rows X to Y from a table?

To display rows 5 to 7, construct a query like this:

```
SELECT *
FROM   tableX
WHERE  rowid in (
   SELECT rowid FROM tableX
   WHERE rownum <= 7
  MINUS
   SELECT rowid FROM tableX
   WHERE rownum < 5);
```

Please note, there is no explicit row order in a relational database. However, this query is quite fun and may even help in the odd situation.

- Back to top of file

How does one select EVERY N th row from a table?

One can easily select all even, odd, or Nth rows from a table using SQL queries like this:

Method 1: Using a subquery

```
SELECT *
FROM   emp
WHERE  (ROWID,0) IN (SELECT ROWID, MOD(ROWNUM,4)
                     FROM   emp);
```

Method 2: Use dynamic views (available from Oracle7.2):

```
SELECT *
FROM   ( SELECT rownum rn, empno, ename
         FROM emp
       ) temp
WHERE  MOD(temp.ROWNUM,4) = 0;
```

Please note, there is no explicit row order in a relational database. However, these queries are quite fun and may even help in the odd situation.

- Back to top of file

How does one select the TOP N rows from a table?

From Oracle8i one can have an inner-query with an ORDER BY clause. Look at this example:

```
SELECT *
FROM   (SELECT * FROM my_table ORDER BY col_name_1 DESC)
WHERE  ROWNUM < 10;
```

Use this workaround with prior releases:

```
SELECT *
  FROM my_table a
 WHERE 10 >= (SELECT COUNT(DISTINCT maxcol)
                FROM my_table b
               WHERE b.maxcol >= a.maxcol)
 ORDER BY maxcol DESC;
```

- Back to top of file

How does one code a tree-structured query?

Tree-structured queries are definitely non-relational (enough to kill Codd and make him roll in his grave). Also, this feature is not often found in other database offerings.

The SCOTT/TIGER database schema contains a table EMP with a self-referencing relation (EMPNO and MGR columns). This table is perfect for tesing and demonstrating tree-structured queries as the MGR column contains the employee number of the "current" employee's boss.

The LEVEL pseudo-column is an indication of how deep in the tree one is. Oracle can handle queries with a depth of up to 255 levels. Look at this example:

```
select  LEVEL, EMPNO, ENAME, MGR
  from  EMP
connect by prior EMPNO = MGR
  start with MGR is NULL;
```

One can produce an indented report by using the level number to substring or lpad() a series of spaces, and concatenate that to the string. Look at this example:

```
select lpad(' ', LEVEL * 2) || ENAME ........
```

One uses the "start with" clause to specify the start of the tree. More than one record can match the starting condition. One disadvantage of having a "connect by prior" clause is that you cannot perform a join to other tables. The "connect by prior" clause is rarely implemented in the other database offerings. Trying to do this programmatically is difficult as one has to do the top level query first, then, for each of the records open a cursor to look for child nodes.

One way of working around this is to use PL/SQL, open the driving cursor with the "connect by prior" statement, and the select matching records from other tables on a row-by-row basis, inserting the results into a temporary table for later retrieval.

- Back to top of file

How does one code a matrix report in SQL?

Look at this example query with sample output:

```
SELECT  *
FROM  (SELECT job,
              sum(decode(deptno,10,sal)) DEPT10,
              sum(decode(deptno,20,sal)) DEPT20,
              sum(decode(deptno,30,sal)) DEPT30,
              sum(decode(deptno,40,sal)) DEPT40
         FROM scott.emp
        GROUP BY job)
ORDER BY 1;

JOB           DEPT10     DEPT20     DEPT30     DEPT40
--------- ---------- ---------- ---------- ----------
ANALYST                    6000
CLERK           1300       1900        950
MANAGER         2450       2975       2850
PRESIDENT       5000
SALESMAN                                5600
```


- Back to top of file

How does one implement IF-THEN-ELSE in a select statement?

The Oracle **decode** function acts like a procedural statement inside an SQL statement to return different values or columns based on the values of other columns in the select statement.

Some examples:

```
select decode(sex, 'M', 'Male',
                   'F', 'Female',
                   'Unknown')
from   employees;

select a, b, decode( abs(a-b), a-b, 'a > b',
                               0,   'a = b',
                                    'a < b')
from  tableX;

select decode( GREATEST(A,B), A, 'A is greater than B', 'B is
greater than A')...
```

Note: The decode function is not ANSI SQL and is rarely implemented in other RDBMS offerings. It is one of the good things about Oracle, but use it sparingly if portability is required.

From Oracle 8i one can also use CASE statements in SQL. Look at this example:

```
        SELECT ename, CASE WHEN sal>1000 THEN 'Over paid' ELSE 'Under
paid' END
        FROM   emp;
```

- Back to top of file

How can one dump/ examine the exact content of a database column?

```
        SELECT DUMP(col1)
        FROM tab1
        WHERE cond1 = val1;

        DUMP(COL1)
        ------------------------------------
        Typ=96 Len=4: 65,66,67,32
```

For this example the type is 96, indicating CHAR, and the last byte in the column is 32, which is the ASCII code for a space. This tells us that this column is blank-padded.

- Back to top of file

Can one drop a column from a table?

From Oracle8i one can DROP a column from a table. Look at this sample script, demonstrating the *ALTER TABLE table_name DROP COLUMN column_name;* command.

With previous releases one can use Joseph S. Testa's DROP COLUMN package that can be downloaded from http://www.oracle-dba.com/ora_scr.htm.

Other workarounds:

```
1. SQL> update t1 set column_to_drop = NULL;
   SQL> rename t1 to t1_base;
   SQL> create view t1 as select <specific columns> from t1_base;

2. SQL> create table t2 as select <specific columns> from t1;
   SQL> drop table t1;
   SQL> rename t2 to t1;
```

- Back to top of file

Can one rename a column in a table?

No, this is listed as Enhancement Request 163519. Some workarounds:

```
1. -- Use a view with correct column names...
   rename t1 to t1_base;
   create view t1 <column list with new name> as select * from t1_base;

2. -- Recreate the table with correct column names...
   create table t2 <column list with new name> as select * from t1;
   drop table t1;
   rename t2 to t1;

3. -- Add a column with a new name and drop an old column...
```

```
alter table t1 add ( newcolame datatype );
update t1 set newcolname=oldcolname;
alter table t1 drop column oldcolname;
```

- Back to top of file

How can I change my Oracle password?

Issue the following SQL command: `ALTER USER <username> IDENTIFIED BY <new_password>` `/`

From Oracle8 you can just type "password" from SQL*Plus, or if you need to change another user's password, type "password user_name".

- Back to top of file

How does one find the next value of a sequence?

Perform an "ALTER SEQUENCE ... NOCACHE" to unload the unused cached sequence numbers from the Oracle library cache. This way, no cached numbers will be lost. If you then select from the USER_SEQUENCES dictionary view, you will see the correct high water mark value that would be returned for the next NEXTVALL call. Afterwards, perform an "ALTER SEQUENCE ... CACHE" to restore caching.

You can use the above technique to prevent sequence number loss before a *SHUTDOWN ABORT*, or any other operation that would cause gaps in sequence values.

- Back to top of file

Workaround for snapshots on tables with LONG columns

You can use the SQL*Plus COPY command instead of snapshots if you need to copy LONG and LONG RAW variables from one location to another. Eg:

```
COPY TO SCOTT/TIGER@REMOTE     -
CREATE IMAGE_TABLE USING       -
       SELECT IMAGE_NO, IMAGE  -
       FROM   IMAGES;
```

Note: If you run Oracle8, convert your LONGs to LOBs, as it can be replicated.

Q: What is the JavaTM 2 Platform, Enterprise Edition (J2EETM)?
JavaTM 2 Platform, Enterprise Edition (J2EETM) is a set of coordinated specifications and practices that together enable solutions for developing, deploying, and managing multi-tier server-centric applications. Building on the JavaTM 2 Platform, Standard Edition (J2SETM), J2EE adds the Capabilities necessary to provide a complete, stable, secure, and fast Java platform To the enterprise level. It provides value by significantly reducing the Cost and complexity of developing and deploying multi-tier solutions, Resulting in services that can be rapidly deployed and easily enhanced.
Q: What are the main benefits of J2EE?
J2EE provides the following:

Faster solutions delivery time to market. J2EE uses "containers" to simplify development. J2EE containers provide for the separation of business logic from resource and lifecycle management, which means that developers can focus on writing business logic -- their value add -- rather than

writing enterprise infrastructure. For example, the Enterprise JavaBeans™ (EJB™) container (implemented by J2EE technology vendors) handles distributed communication, threading, scaling, transaction management, etc.
Similarly, Java Servlets simplify web development by providing infrastructure for component, communication, and session management in a web container that is integrated with a web server.

Freedom of choice. J2EE technology is a set of standards that many vendors can implement. The vendors are free to compete on implementations but not on standards or APIs. Sun supplies a comprehensive J2EE Compatibility Test Suite (CTS) to J2EE licensees. The J2EE CTS helps ensure compatibility among the application vendors which helps ensure portability for the applications and components written for J2EE. J2EE brings Write Once, Run Anywhere™ (WORA™) to the server.

Simplified connectivity. J2EE technology makes it easier to connect the applications and systems you already have and bring those capabilities to the web, to cell phones, and to devices. J2EE offers Java Message Service for integrating diverse applications in a loosely coupled, asynchronous way.
J2EE also offers CORBA support for tightly linking systems through remote method calls. In addition, J2EE 1.3 adds J2EE Connectors for linking to enterprise information systems such as ERP systems, packaged financial applications, and CRM applications.

By offering one platform with faster solution delivery time to market, freedom of choice, and simplified connectivity, J2EE helps IT by reducing TCO and simultaneously avoiding single-source for their enterprise software needs.

Q: What technologies are included in J2EE?

The primary technologies in J2EE are: Enterprise JavaBeans™ (EJBs™), JavaServer Pages™ (JSPs™), Java Servlets, the Java Naming and Directory Interface™ (JNDI™), the Java Transaction API (JTA), CORBA, and the JDBC™ data access API.

Q: How does J2EE relate to Enterprise JavaBeans technology?

Enterprise JavaBeans (EJB) technology is the basis of J2EE. EJB technology provides the scalable architecture for executing business logic in a

distributed computing environment. J2EE makes the life of an enterprise developer easier by combining the EJB component architecture with other enterprise technologies to solutions on the Java platform for seamless development and deployment of server side applications.

Q: Who needs J2EE?

ISVs need J2EE because it gives them a blueprint for providing a complete enterprise computing solution on the Java platform. Enterprise developers need J2EE because writing distributed business applications is hard, and they need a high-productivity solution that allows them to focus only on writing their business logic and having a full range of enterprise-class services to rely on, like transactional distributed objects, message oriented middleware, and naming and directory services.