Answers

1) B
2) C
3) A
4) D
5) Non-static members
6) Static members
7) Default
8) abstract
9) Local variables
10) Data type default value
11) Data type default value
12) No
13) No
14) Yes
15) No
16) No
17) Not to instantiate
18) No inheritance takes place
19) Choosing which class method want to run at runtime
20) Abstraction
21) Pass-by-reference
22) To expose class implemented methods as interface methods
23) Using this keyword
24) No
25) Data type default value
26) Default
27) Yes
28) No, only can contain non-static methods
29) By creating objects to static inner class
30) Using “length” property
31) Using length() method
32) No
33) Yes
34) Static variables
35) When class loads into JVM
36) No
37) They are executed before every constructor execution
38) Constructor block
39) Class cannot be inherited and instantiated

40) Class can be inherited and instantiated with the package
41) Can be accessible anywhere in the package and only up to sub classes outside the package
42) Can accessible however we want in both within the package and outside the package
43) a, b, f
44) 9
45) IndexArrayOutOfBoundsException raises
46) CheckedException – InterruptedException
47) RuntimeException sub classes and direct Exception sub classes such as InterruptedException, FileNotFoundException, IOException etc
48) ArithmeticException, NumberFormatException
49) CheckedExceptions
50) Program abnormally terminates
51) Using System.exit(0)
52) Loss of precision error comes
53) append() and insert()
54) Yes, char as parameter
55) Its size doubles
56) By passing capacity as argument in StringBuffer class constructor, and ensureCapacity() method
57) Remaining characters will be discarded
58) Using trim() method
59) Their address is compared
60) Their state is compared
61) == operator
62) IS-A is used in inheritance and HAS-A is used in instantiation
63) No
64) Yes, within the package
65) Yes, up to sub class outside the package
66) Yes, anywhere in “within the package” and “outside the package”
67) Usually public class names are in package, they can be accessible anywhere in the system, such classes must not be overridden by default package classes
68) No
69) Yes
70) Yes
71) Yes
72) Write EmpPayroll, Printer class examples
73) Goto, return, private, protected, public, abstract, final, static, transient, volatile, native, synchronized, continue, break, class, package, switch, case, default, instanceof, for, while, if, throw, throws, finally
74) Describe all access specifiers and modifiers briefly
    a. Private – members allowed within the class
    b. default – members allowed within the package
    c. protected – members allowed up to sub class outside the package

d. public – anywhere in the system
e. abstract – methods are declared but not implemented
f. final – variables cannot be initialized, methods cannot be overridden
g. static – members are not having instance memory
h. transient – variables that cannot be serialized
i. volatile – variables that cannot be synchronized
j. native – methods for which implementation exist in C language
k. synchronized – only one thread can access method at a time

75) Arithmetic Operators: +, -, *, /, %
76) Relational Operators: <, >, <=, >=, ==, !=
77) Bitwise AND (&), bitwise OR (|), bitwise XOR (^), bitwise compliment (~) operators
78) Logical Operators: logical AND (&&), logical OR (||), logical NOT (!)
79) Assignment Operators: simple assignment operator (=), Add AND assign (+=), Subtract AND assign (-=), Multiply AND assign (*=), Divide AND assign (/=), Modulus AND assign (%=)
80) Prefix operator Pre/Post increment (++), Pre/Post decrement (- -)
81) Shift Operators: Left shift (<<), Right shift (>>), Right shift zero fill (>>>) operators
82) Ternary Operator: (condition)?lhs_value:rhs_value
83) In switch statements and break
84) We must extend from Thread super class or Runnable interface.
85) public void run() method
86) Thread.sleep()
87) setName(String name)
88) getName()
89) setPriority(int priority)
90) MIN_PRIORITY (1), NORM_PRIORITY (5), MAX_PRIORITY (10)
91) Synchronized
92) By applying Synchronized on method
93) By keeping object in Synchronized block
94) It takes full priority until its execution completes
95) Based on number of methods want to run concurrently
96) read()
97) write()
98) FileInputStream
99) Int value, the ASCII/Unicode charcter value of characters read from InputStream
100) -1
101) DataInputStream
102) BufferedInputStream
103) PrintStream
104) PrintStream class object “out”
105) InputStream and OutputStream classes deals with stream of characters, Reader & Writer classes deals with stream of charaters
106) ServerSocket class

107) Socket class
108) accept() method
109) DatagramSocket and DatagramPacket
110) Vector, Stack, Hashtable, Properties
111) Dictionary
112) Hashtable
113) Hashtable accept key and values as Objects and Properties class accepts Strings as object and values
114) add(Object o), add(int index, Object o), set(int index, Object o), get(int index), getObject(o), remove(int index), remove(Object o), Enumeration elements()
115) Stack
116) FIFO
117) LIFO
118) elements()
119) Enumeration
120) All legacy class methods are synchronized
121) Convenient methods are given in Legacy classes that internally use indexes. They automatically grow and shrink, they accepts objects of different data types.
122) add(Object o), add(int index, Object o), set(int index, Object o), get(int index), getObject(o), remove(int index), remove(Object o), Enumeration elements()
123) put(Object key, Object value), get(Object key), remove(Object key,) Enumeration keys()
124) setProperty(String name, String value), getProperty(String name), remove(Object key), Set<String> stringPropertyNames()
125) Iterable
126) List and Set
127) ArrayList
128) Iterator iterator ()
129) ListIterator listIterator ()
130) List in Ordered Collection, Set is Unique Collection
131) List interface methods takes index as argument, Set interface doesn't take index argument
132) ArrayList, LinkedList
133) HashSet, TreeSet, LinkedHashSet
134) HashSet maintains Unique-no order , TreeSet maintains Unique-natural order, LinkedHashSet maintains Unique-given order
135) HashMap, TreeMap, LinkedHashMap
136) HashMap maintains Unique-no order , TreeMap maintains Unique-natural order, LinkedHashMap maintains Unique-given order
137) Java.lang.Observable and java.util.Observer
138) Comparable interface method is int compareTo(Object o), Comparator interface method is compare(Object lhs, Object rhs)

139) Legacy class methods are synchronized, Collection class methods are asynchronous, Legacy class methods not inter-operable with C++ language STL (Standard Template Library) classes, but Collection class methods does.

140) LinkedList class

a. peek() retrieve and doesn't remove head element, returns null if queue is empty
b. element() retrieve and doesn't remove head element, throws NoSuchElementException if queue is empty
c. poll() method retrieve and remove head element
d. remove() retrieve and removes head element, throws NoSuchElementException if queue is empty

141) peekLast() retrieve and don't remove last element and pollLast() retrieve and remove last element, both will return null if list is empty.

142) peek() and poll() methods belongs to LinkedList.

143) If Thread class run() method is directly called it runs as a normal method.

144) Thread class start() method runs run() method

145) The addShutdownHook(Thread hook) method of Runtime class is used to register the thread with the Virtual Machine.

```
Runtime.getRuntime().addShutdownHook(new Thread() {
        public void run() {
          System.out.println("JVM is exiting");
        }
});
```

146)