

# **CORE JAVA**

## **With**

# **SCJP / OCJP**

## **Question Bank**



**DURGA M.Tech**

**(Sun certified & Realtime Expert)**

**Ex. IBM Employee**

**Trained Lakhs of Students  
for last 14 years across INDIA**

**India's No.1 Software Training Institute**

# **DURGASOFT**

**[www.durgasoft.com](http://www.durgasoft.com) Ph: 9246212143, 8096969696**

SNO	Table of Contents	PageNo
1	Language Fundamentals	3 - 20
2	Declarations and Access Control	21 - 41
3	Operators	42 - 48
4	Flow Control	49 - 55
5	Exception Handling	56 - 69
6	Asserions	70 - 76
7	Garbage Collections	77 - 82
8	OOPs	83 - 140
9	Multi-Threading	141 - 167
10	Java.lang Package	168 - 178
11	java.io package and Serialization	179 - 193
12	Collections and Generics	194 - 231
13	Inner Classes	232 - 234
14	Internationalization	235 - 243
15	Development	244 - 249
16	<b>DURGA's INTERVIEW QUESTIONS</b>	250 - 263

# 1. Language Fundamentals

Q: 1 Given

10. class Foo {

11. static void alpha() { /\* more code here \*/ }

12. void beta() { /\* more code here \*/ }

13. }

Page 3 of 263

Which two statements are true? (Choose two.)

A. Foo.beta() is a valid invocation of beta().

B. Foo.alpha() is a valid invocation of alpha().

C. Method beta() can directly call method alpha().

D. Method alpha() can directly call method beta().

Answer: B, C

Q: 2 Given:

12. public class Yippee2 {

13.

14. static public void main(String [] yahoo) {

15. for(int x = 1; x < yahoo.length; x++) {

16. System.out.print(yahoo[x] + " ");

17. }

18. }

19. }

and the command line invocation:

java Yippee2 a b c

**What is the result?**

- A. a b
- B. b c
- C. a b c
- D. Compilation fails.
- E. An exception is thrown at runtime.

**Answer: B**

**Q: 3 Given:**

```
15. public class Yippee {  
16.     public static void main(String [] args) {  
17.         for(int x = 1; x < args.length; x++) {  
18.             System.out.print(args[x] + " ");  
19.         }  
20.     }  
21. }
```

**and two separate command line invocations:**

**java Yippee**

**java Yippee 1 2 3 4**

**What is the result?**

- A. No output is produced.  
1 2 3
- B. No output is produced.  
2 3 4
- C. No output is produced.  
1 2 3 4

D. An exception is thrown at runtime.

1 2 3

E. An exception is thrown at runtime.

2 3 4

F. An exception is thrown at runtime.

1 2 3 4

**Answer: B**

**Q: 4 Given a class Repetition:**

1. package utils;

2.

3. public class Repetition {

4. public static String twice(String s) { return s + s; }

5. }

and given another class Demo:

1. // insert code here

2.

3. public class Demo {

4. public static void main(String[] args) {

5. System.out.println(twice("pizza"));

6. }

7. }

**Which code should be inserted at line 1 of Demo.java to compile and run Demo to print "pizzapizza"?**

A. import utils.\*;

B. static import utils.\*;

C. import utils.Repetition.\*;

- D. static import utils.Repetition.\*;
- E. import utils.Repetition.twice();
- F. import static utils.Repetition.twice;
- G. static import utils.Repetition.twice;

**Answer: F**

**Q: 5 A JavaBeans component has the following field:**

**11. private boolean enabled;**

**Which two pairs of method declarations follow the JavaBeans standard for accessing this field?  
(Choose two.)**

- A. public void setEnabled( boolean enabled )  
public boolean getEnabled()
- B. public void setEnabled( boolean enabled )  
public void isEnabled()
- C. public void setEnabled( boolean enabled )  
public boolean isEnabled()
- D. public boolean setEnabled( boolean enabled )  
public boolean getEnabled()

**Answer: A, C**

**Q: 6**

**Given classes defined in two different files:**

- 1. package util;
- 2. public class BitUtils {
- 3. public static void process(byte[]) { /\* more code here \*/ }
- 4. }

```
1. package app;
2. public class SomeApp {
3.     public static void main(String[] args) {
4.         byte[] bytes = new byte[256];
5.         // insert code here
6.     }
7. }
```

**What is required at line 5 in class SomeApp to use the process method of BitUtils?**

- A. process(bytes);
- B. BitUtils.process(bytes);
- C. util.BitUtils.process(bytes);
- D. SomeApp cannot use methods in BitUtils.
- E. import util.BitUtils.\*; process(bytes);

**Answer: C**

**Q: 7 Given:**

```
enum Example { ONE, TWO, THREE }
```

**Which statement is true?**

- A. The expressions (ONE == ONE) and ONE.equals(ONE) are both guaranteed to be true.
- B. The expression (ONE < TWO) is guaranteed to be true and ONE.compareTo(TWO) is guaranteed to be less than one.
- C. The Example values cannot be used in a raw java.util.HashMap; instead, the programmer must use a java.util.EnumMap.
- D. The Example values can be used in a java.util.SortedSet, but the set will NOT be sorted because enumerated types do NOT implement java.lang.Comparable.

**Answer: A**

Q: 8 Given:

```
11. public abstract class Shape {  
12.     private int x;  
13.     private int y;  
14.     public abstract void draw();  
15.     public void setAnchor(int x, int y) {  
16.         this.x = x;  
17.         this.y = y;  
18.     }  
19. }
```

Which two classes use the Shape class correctly? (Choose two.)

A. public class Circle implements Shape {  
private int radius;  
}

B. public abstract class Circle extends Shape {  
private int radius;  
}

C. public class Circle extends Shape {  
private int radius;  
public void draw();  
}

D. public abstract class Circle implements Shape {  
private int radius;  
public void draw();



}

E. public class Circle extends Shape {

private int radius;

public void draw() { /\* code here \*/ }

F. public abstract class Circle implements Shape {

private int radius;

public void draw() { /\* code here \*/ }

**Answer: B, E**

**Q: 09 Given:**

10. class Nav{

11. public enum Direction { NORTH, SOUTH, EAST, WEST }

12. }

13. public class Sprite{

14. // insert code here

15. }

**Which code, inserted at line 14, allows the Sprite class to compile?**

A. Direction d = NORTH;

B. Nav.Direction d = NORTH;

C. Direction d = Direction.NORTH;

D. Nav.Direction d = Nav.Direction.NORTH;

**Answer: D**

**Q: 10 Click the Exhibit button.**

**Which three statements are true? (Choose three.)**

```
10. interface Foo {
11.     int bar();
12. }
13.
14. public class Beta {
15.
16.     class A implements Foo {
17.         public int bar() { return 1; }
18.     }
19.
20.     public int fubar( Foo foo ) { return
foo.bar(); }
21.
22.     public void testFoo() {
23.
24.         class A implements Foo {
25.             public int bar() { return 2; }
26.         }
27.
28.         System.out.println( fubar( new A() )
);
29.     }
30.
31.     public static void main( String[] argv
) {
32.         new Beta().testFoo();
33.     }
34. }
```

- A. Compilation fails.
- B. The code compiles and the output is 2.
- C. If lines 16, 17 and 18 were removed, compilation would fail.
- D. If lines 24, 25 and 26 were removed, compilation would fail.
- E. If lines 16, 17 and 18 were removed, the code would compile and the output would be 2.
- F. If lines 24, 25 and 26 were removed, the code would compile and the output would be 1.

**Answer: B, E, F**

**Q: 11 Click the Task button.**

Place the code fragments in position to complete the Displayable interface.

```
interface Reloadable {
    public void reload();
}

class Edit {
    public void edit() { /* Edit Here */ }
}

interface Displayable
    _____
    _____
    _____
}
```

#### Code Fragments

extends	public void display();	Reloadable
implements	public void display() { /* Display Here */ }	Edit

#### Solution:

```
interface Reloadable{

    public void reload();

}

class Edit{

    public void edit(){/* Edit Here*/}

}

interface Displayable

    extends Reloadable {

        public void display();

    }
```

#### Q:12 Given:

35. String #name = "Jane Doe";

36. int \$age = 24;

37. Double \_height = 123.5;

38. double ~temp = 37.5;

**Which two statements are true? (Choose two.)**

- A. Line 35 will not compile.
- B. Line 36 will not compile.
- C. Line 37 will not compile.
- D. Line 38 will not compile.

**Answer: A, D**

**Q: 13 Given:**

55. `int [] x = {1, 2, 3, 4, 5};`

56. `int y[] = x;`

57. `System.out.println(y[2]);`

**Which statement is true?**

- A. Line 57 will print the value 2.
- B. Line 57 will print the value 3.
- C. Compilation will fail because of an error in line 55.
- D. Compilation will fail because of an error in line 56.

**Answer: B**

**Q: 14**

**A programmer needs to create a logging method that can accept an arbitrary number of arguments. For example, it may be called in these ways:**

`logIt("log message1");`

`logIt("log message2", "log message3");`

`logIt("log message4", "log message5", "log message6");`

**Which declaration satisfies this requirement?**

- A. `public void logIt(String * msgs)`
- B. `public void logIt(String [] msgs)`
- C. `public void logIt(String... msgs)`
- D. `public void logIt(String msg1, String msg2, String msg3)`

**Answer: C**

**Q: 15**

**Which two code fragments correctly create and initialize a static array of int elements? (Choose two.)**

- A. `static final int[] a = { 100,200 };`
- B. `static final int[] a;`  
`static { a=new int[2]; a[0]=100; a[1]=200; }`
- C. `static final int[] a = new int[2]{ 100,200 };`
- D. `static final int[] a;`  
`static void init() { a = new int[3]; a[0]=100; a[1]=200; }`

**Answer: A, B**

**Q: 16 Given:**

- 11. `public static void main(String[] args) {`
- 12. `String str = "null";`
- 13. `if (str == null) {`
- 14. `System.out.println("null");`
- 15. `} else (str.length() == 0) {`
- 16. `System.out.println("zero");`
- 17. `} else {`
- 18. `System.out.println("some");`
- 19. `}`

20. }

**What is the result?**

- A. null
- B. zero
- C. some
- D. Compilation fails.
- E. An exception is thrown at runtime.

**Answer: D**

**Q: 17 Click the Exhibit button.**

**Given:**

**34. Test t = new Test();**

**35. t.method(5);**

**What is the output from line 5 of the Test class?**

```
1. public class Test {  
2.     int x = 12;  
3.     public void method(int x) {  
4.         x+=x;  
5.         System.out.println(x);  
6.     }  
7. }
```

- A. 5
- B. 10
- C. 12
- D. 17
- E. 24

**Answer: B**

**Q: 18 Given**

```
11. public interface Status {  
12. /* insert code here */ int MY_VALUE = 10;  
13. }
```

**Which three are valid on line 12? (Choose three.)**

- A. final
- B. static
- C. native
- D. public
- E. private
- F. abstract
- G. protected

**Answer: A, B, D**

**Question: 19**

A programmer is designing a class to encapsulate the information about an inventory item. A JavaBeans component is needed to do this. The InventoryItem class has private instance variables to store the item information:

- 10. private int itemId;
- 11. private String name;
- 12. private String description;

**Which method signature follows the JavaBeans naming standards for modifying the itemId instance variable?**

- A. itemID(int itemId)

- B. update(int itemId)
- C. setItemId(int itemId)
- D. mutatelItemId(int itemId)
- E. updateltemlD(int itemId)

**Answer: C**

**Question:20**

**Given a file GrizzlyBear.java:**

1. package animals.mammals;
- 2.
3. public class GrizzlyBear extends Bear {
4. void hunt() {
5. Salmon s = findSalmon();
6. s.consume();
7. }
8. }

**and another file, Salmon.java:**

1. package animals.fish;
- 2.
3. public class Salmon extends Fish {
4. void consume() { /\* do stuff \*/ }
5. }

**Assume both classes are defined in the correct directories for theft packages, and that the Mammal class correctly defines the findSalmon() method. Which two changes allow this code to compile correctly? (Choose two.)**



- A. add public to the start of line 4 in Salmon.java
- B. add public to the start of line 4 in GrizzlyBear.java
- C. add import animals.mammals.\*; at line 2 in Salmon.java
- D. add import animals.fish.\*; at line 2 in GrizzlyBear.java
- E. add import animals.fish.Salmon.\*; at line 2 in GrizzlyBear.java
- F. add import animals.mammals.GrizzlyBear.\*;at line 2 in Salmon.java

**Answer: AD**

**Question:21**

**Given:**

- 11. public class Barn {
- 12. public static void main(String[] args) {
- 13. new Barn().go("hi", 1);
- 14. new Barn().go("hi", "world", 2);
- 15. }
- 16. public void go(String... y, int x) {
- 17. System.out.print(y[y.length - 1] + " ");
- 18. }
- 19. }

**What is the result?**

- A. hi hi
- B. hi world
- C. world world
- D. Compilation fails.
- E. An exception is thrown at runtime.

**Answer: D**

**Question:22**

**Given:**

```
11. class Mud {  
12. // insert code here  
13. 14. }  
  
System.out.println("hi");  
  
15. }
```

**And the following five fragments:**

```
public static void main(String...a) {  
public static void main(String.* a) {  
public static void main(String... a) {  
public static void main(String[]... a) {  
public static void main(String...[] a) {
```

**How many of the code fragments, inserted independently at line 12, compile?**

- A. 0
- B. 1
- C. 2
- D. 3
- E. 4
- F. 5

**Answer: D**

**Question:23**

**Given:**

```
1. class Alligator {  
2. public static void main(String[] args) {  
3. int []x[] = {{1,2}, {3,4,5}, {6,7,8,9}};
```

4. `int [][]y = x;`

5. `System.out.println(y[2][1]);`

6. `}`

7. `}`

What is the result?

A. 2

B. 3

C. 4

D. 6

E. 7

F. Compilation fails.

**Answer:** E

**Question:24**

1. `public class Venus {`

2. `public static void main(String[] args) {`

3. `int [] x = {1,2,3};`

4. `int y[] = {4,5,6};`

5. `new Venus().go(x,y);`

6. `}`

7. `void go(int[]... z) {`

8. `for(int[] a : z)`

9. `System.out.print(a[0]);`

10. `}`

11. `}` What is the result?

A. 1

B. 12

- C. 14
- D. 123
- E. Compilation fails.
- F. An exception is thrown at runtime.

**Answer:** C

**Question:25**

**Which two code fragments correctly create and initialize a static array of int elements?**

**(Choose two.)**

- A. `static final int[] a = { 100,200 };`
- B. `static final int[] a; static { a=new int[2]; a[0]=100; a[1]=200; }`
- C. `static final int[] a = new int[2]{ 100,200 };`
- D. `static final int[] a; static void init() { a = new int[3]; a[0]=100; a[1]=200; }`

**Answer:** A,B

## 2. Declarations and Access Control

Q: 1 Click the Task button.

Place the lines in the correct order to complete the enum.

```
enum Element {
```

1st

2nd

3rd

4th

5th

Lines

```
public String info() { return "element"; }
```

```
};
```

```
FIRE { public String info() { return "Hot"; } }
```

```
EARTH, WIND,
```

```
}
```

Solution:

```
enum Element{  
    EARTH,WIND,  
    FIRE{public String info(){return "Hot";}  
    };  
    public String info(){return "element";}  
}
```

Q: 2 Given:

10. `package com.sun.scjp;`

11. `public class Geodetics {`

12. `public static final double DIAMETER = 12756.32; // kilometers`

13. `}`

**Which two correctly access the DIAMETER member of the Geodetics class? (Choose two.)**

A. `import com.sun.scjp.Geodetics;`

`public class TerraCarta {`

`public double halfway()`

`{ return Geodetics.DIAMETER/2.0; }`

B. `import static com.sun.scjp.Geodetics;`

`public class TerraCarta{`

`public double halfway() { return DIAMETER/2.0; } }`

C. `import static com.sun.scjp.Geodetics.*;`

`public class TerraCarta {`

`public double halfway() { return DIAMETER/2.0; } }`

D. `package com.sun.scjp;`

`public class TerraCarta {`

`public double halfway() { return DIAMETER/2.0; } }`

**Answer: A, C**

**Q: 3 Click the Task button.**

Place the code elements in order so that the resulting Java source file will compile correctly, resulting in a class called com.sun.cert.AddressBook.

Source File	Code Element
1st	package com.sun.cert;
2nd	package com.sun.cert.*;
3rd	import java.util.*;
	import java.*;
ArrayList entries;	public class AddressBook {
}	public static class AddressBook {

Done

**Solution:**

```
package com.sun.cert;

import java.util.*;

public class AddressBook{

    ArrayList entries;

}
```

**Q: 4 Which two classes correctly implement both the java.lang.Runnable and the java.lang.Cloneable interfaces? (Choose**

```
A. public class Session

implements Runnable, Cloneable {

public void run();

public Object clone();

}
```

B. public class Session

extends Runnable, Clonable {

public void run() { /\* do something \*/ }

public Object clone() { /\* make a copy \*/ }

C. public class Session

implements Runnable, Clonable {

public void run() { /\* do something \*/ }

public Object clone() { /\* make a copy \*/ }

D. public abstract class Session

implements Runnable, Clonable {

public void run() { /\* do something \*/ }

public Object clone() { /\*make a copy \*/ }

E. public class Session

implements Runnable, implements Clonable {

public void run() { /\* do something \*/ }

public Object clone() { /\* make a copy \*/ }

**Answer: C, D**

**Q: 5 Given classes defined in two different files:**

1. package util;

2. public class BitUtils {

3. private static void process(byte[] b) {}

4. }

1. package app;

2. public class SomeApp {

3. public static void main(String[] args) {



4. `byte[] bytes = new byte[256];`

5. `// insert code here`

6. `}`

7. `}`

**What is required at line 5 in class `SomeApp` to use the `process` method of `BitUtils`?**

- A. `process(bytes);`
- B. `BitUtils.process(bytes);`
- C. `app.BitUtils.process(bytes);`
- D. `util.BitUtils.process(bytes);`
- E. `import util.BitUtils.*; process(bytes);`
- F. `SomeApp` cannot use the `process` method in `BitUtils`.

**Answer: F**

**Q: 6 Given:**

11. `class Cup { }`

12. `class PoisonCup extends Cup { }`

...

21. `public void takeCup(Cup c) {`

22. `if (c instanceof PoisonCup) {`

23. `System.out.println("Inconceivable!");`

24. `} else if (c instanceof Cup) {`

25. `System.out.println("Dizzying intellect!");`

26. `} else {`

27. `System.exit(0);`

28. `}`

29. `}`

And the execution of the statements:

```
Cup cup = new PoisonCup();
```

```
takeCup(cup);
```

What is the output?

- A. Inconceivable!
- B. Dizzying intellect!
- C. The code runs with no output.
- D. An exception is thrown at runtime.
- E. Compilation fails because of an error in line 22.

Answer: A

Q: 7 Click the Exhibit button.

```
public class A
{
    private int counter=0;

    public static int getInstanceCount()
    {
        return counter;
    }

    public A()
    {
        counter++;
    }
}
```

Given this code from Class B:

25. A a1 = new A();

26. `A a2 = new A();`

27. `A a3 = new A();`

28. `System.out.println(A.getInstanceCount());`

**What is the result?**

- A. Compilation of class A fails.
- B. Line 28 prints the value 3 to System.out.
- C. Line 28 prints the value 1 to System.out.
- D. A runtime error occurs when line 25 executes.
- E. Compilation fails because of an error on line 28.

**Answer: A**

**Q:8 Given:**

11. `String[] elements = { "for", "tea", "too" };`

12. `String first = (elements.length > 0) ? elements[0] : null;`

**What is the result?**

- A. Compilation fails.
- B. An exception is thrown at runtime.
- C. The variable first is set to null.
- D. The variable first is set to elements[0].

**Answer: D**

**Q:09 Given:**

11. `interface DeclareStuff {`

12. `public static final int EASY = 3;`

13. `void doStuff(int t); }`

```
14. public class TestDeclare implements DeclareStuff {  
15. public static void main(String [] args) {  
16. int x = 5;  
17. new TestDeclare().doStuff(++x);  
18. }  
19. void doStuff(int s) {  
20. s += EASY + ++s;  
21. System.out.println("s " + s);  
22. }  
23. }
```

What is the result?

- A. s 14
- B. s 16
- C. s 10
- D. Compilation fails.
- E. An exception is thrown at runtime.

**Answer: D**

Q: 10 Given:

```
1. public class TestString1 {  
2. public static void main(String[] args) {  
3. String str = "420";  
4. str += 42;  
5. System.out.print(str);  
6. }  
7. }
```

**What is the output?**

- A. 42
- B. 420
- C. 462
- D. 42042
- E. Compilation fails.
- F. An exception is thrown at runtime.

**Answer: D**

**Q: 11 Given:**

```
11. class Converter {  
12. public static void main(String[] args) {  
13. Integer i = args[0];  
14. int j = 12;  
15. System.out.println("It is " + (j==i) + " that j==i.");  
16. }  
17. }
```

**What is the result when the programmer attempts to compile the code and run it with the command `java Converter 12`?**

- A. It is true that `j==i`.
- B. It is false that `j==i`.
- C. An exception is thrown at runtime.
- D. Compilation fails because of an error in line 13.

**Answer: D**

**Q: 12 Given:**

```
10. int x = 0;  
11. int y = 10;
```

```
12. do {  
13. y--;  
14. ++x;  
15. } while (x < 5);  
16. System.out.print(x + "," + y);
```

What is the result?

- A. 5,6
- B. 5,5
- C. 6,5
- D. 6,6

**Answer: B**

**Q: 13 Given:**

```
1. public interface A {  
2. String DEFAULT_GREETING = "Hello World";  
3. public void method1();  
4. }
```

**A programmer wants to create an interface called B that has A as its parent. Which interface declaration is correct?**

- A. public interface B extends A {}
- B. public interface B implements A {}
- C. public interface B instanceof A {}
- D. public interface B inheritsFrom A {}

**Answer: A**

**Q: 14 Given:**

```
11. public enum Title {  
12. MR("Mr."), MRS("Mrs."), MS("Ms.");  
13. private final String title;
```

```
14. private Title(String t) { title = t; }  
15. public String format(String last, String first) {  
16. return title + " " + first + " " + last;  
17. }  
18. }  
19. public static void main(String[] args) {  
20. System.out.println(Title.MR.format("Doe", "John"));  
21. }
```

**What is the result?**

- A. Mr. John Doe
- B. An exception is thrown at runtime.
- C. Compilation fails because of an error in line 12.
- D. Compilation fails because of an error in line 15.
- E. Compilation fails because of an error in line 20.

**Answer: A**

**Q: 15 Given:**

```
1. package test;  
2.  
3. class Target {  
4. public String name = "hello";  
5. }
```

**What can directly access and change the value of the variable name?**

- A. any class
- B. only the Target class
- C. any class in the test package
- D. any class that extends Target

**Answer: C**

Q: 16 Given:

```
11. public class Ball{  
12. public enum Color { RED, GREEN, BLUE };  
13. public void foo(){  
14. // insert code here  
15. { System.out.println(c); }  
16. }  
17. }
```

Which code inserted at line 14 causes the foo method to print RED, GREEN, and BLUE?

- A. for( Color c : Color.values() )
- B. for( Color c = RED; c <= BLUE; c++ )
- C. for( Color c ; c.hasNext() ; c.next() )
- D. for( Color c = Color[0]; c <= Color[2]; c++ )
- E. for( Color c = Color.RED; c <= Color.BLUE; c++ )

**Answer: A**

Q: 17 Click the Task button.

Insert six modifiers into the code such that it meets all of these requirements:

1. It must be possible to create instances of Alpha and Beta from outside the packages in which they are defined.
2. When an object of type Alpha (or any potential subclass of Alpha) has been created, the instance variable alpha may never be changed.
3. The value of the instance variable alpha must always be "A" for objects of type Alpha.

**Code**

```
package alpha;  
Place here class Alpha {  
    Place here String alpha;  
    Place here Alpha() { this("A"); }  
    Place here Alpha(String a) { alpha = a; }  
}  
  
package beta;  
Place here class Beta extends alpha.Alpha {  
    Place here Beta(String a) { super(a); }  
}
```

**Modifiers**

private  
protected  
public

Done



**Solution:**

```
package alpha;

public class Alpha{

    private String alpha;

    public Alpha(){ this("A") ; }

    protected Alpha(String a){ alpha=a; }

}

package beta;

public class Beta extends alpha.Alpha{

    private Beta(String a){ super(a); }

}
```

**Q: 18 Given:**

1. public class Target {
2. private int i = 0;
3. public int addOne(){
4. return ++i;
5. }
6. }

**And:**

1. public class Client {
2. public static void main(String[] args){
3. System.out.println(new Target().addOne());
4. }
5. }

**Which change can you make to Target without affecting Client?**

- A. Line 4 of class Target can be changed to return i++;

- B. Line 2 of class Target can be changed to private int i = 1;
- C. Line 3 of class Target can be changed to private int addOne(){
- D. Line 2 of class Target can be changed to private Integer i = 0;

**Answer: D**

**Q: 19 Click the Task button.**

Replace two of the Modifiers that appear in the Single class to make the code compile.  
Note: Three modifiers will not be used and four modifiers in the code will remain unchanged.

**Code**

```
public class Single {
    private static Single instance;
    public static Single getInstance() {
        if (instance == null) instance = create();
        return instance;
    }
    private Single() { }
    protected Single create() { return new Single(); }
}
class SingleSub extends Single {
}
```

**Modifiers**

final  
protected  
private  
abstract  
static

Done

**Solution:**

```
public class Single{
    private static Single instance;

    public static Single getInstance(){
        if(instance==null) instance = create();
        return instance;
    }
    protected Single() { }
    static Single create () { return new Single () ; }
}
class SingleSub extends Shape{
```

}

Q: 20 Given:

```
12. public class Test {  
13.     public enum Dogs {collie, harrier};  
14.     public static void main(String [] args) {  
15.         Dogs myDog = Dogs.collie;  
16.         switch (myDog) {  
17.             case collie:  
18.                 System.out.print("collie ");  
19.             case harrier:  
20.                 System.out.print("harrier ");  
21.         }  
22.     }  
23. }
```

What is the result?

- A. collie
- B. harrier
- C. Compilation fails.
- D. collie harrier
- E. An exception is thrown at runtime.

Answer: D

Q: 21 Click the Exhibit button.

Given:

```
ClassA a = new ClassA();  
a.methodA();
```

What is the result?

```
10. public class ClassA {
11.     public void methodA() {
12.         ClassB classB = new ClassB();
13.         classB.getValue();
14.     }
15. }
```

And:

```
20. class ClassB {
21.     public ClassC classC;
22.
23.     public String getValue() {
24.         return classC.getValue();
25.     }
26. }
```

And:

```
30. class ClassC {
31.     public String value;
32.
33.     public String getValue() {
34.         value = "ClassB";
35.         return value;
36.     }
37. }
```

- A. Compilation fails.
- B. ClassC is displayed.
- C. The code runs with no output.
- D. An exception is thrown at runtime.

**Answer: D**

Q: 22 Click the Task button.

Place code fragments into position so the output is: The quantity is 420

```

    Place here    update(int quantity, int adjust) {
        Place here
    }

    public void callUpdate() {
        int quant = 100;
        Place here
        System.out.println("The quantity is " + quant);
    }

```

**Code Fragments**

public int	quantity = quantity + adjust;	update(quant, 320);
public void	quant = update(quant, 320);	quantity = quantity + adjust; return quantity;

**Solution:**

```

public int update(int quantity,int adjust){
    quantity=quantity+adjust;
    return quantity;
}

public void call Update() {
    int quant=100;
    quant=update(quant,320);
    System.out.println("the quantity is " +quant);
}

```

Q: 23 Given:

1. package sun.scjp;
2. public enum Color { RED, GREEN, BLUE }
1. package sun.beta;
2. // insert code here

```
3. public class Beta {  
4. Color g = GREEN;  
5. public static void main( String[] argv)  
6. { System.out.println( GREEN); }  
7. }
```

The class Beta and the enum Color are in different packages.

Which two code fragments, inserted individually at line 2 of the Beta declaration, will allow this code to compile? (Choose two.)

- A. import sun.scjp.Color.\*;
- B. import static sun.scjp.Color.\*;
- C. import sun.scjp.Color; import static sun.scjp.Color.\*;
- D. import sun.scjp.\*; import static sun.scjp.Color.\*;
- E. import sun.scjp.Color; import static sun.scjp.Color.GREEN;

Answer: CE

#### Question 24

Given:

```
10. public class Fabric  
11. public enum Color {  
12. RED(0xff0000), GREEN(0x00ff00), BLUE(0x0000ff);  
13. private final int rgb;  
14. Color( int rgb) { this.rgb = rgb; }  
15. public int getRGB() { return rgb; }
```

```
16. };  
17. public static void main( String[] argv) {  
18. // insert code here  
19. }  
20. }
```

Which two code fragments, inserted independently at line 18, allow the Fabric class to compile? (Choose two.)

- A. Color skyColor = BLUE;
- B. Color treeColor = Color.GREEN;
- C. Color purple = new Color( 0xff00ff);
- D. if( RED.getRGB() < BLUE.getRGB() ) {}
- E. Color purple = Color.BLUE + Color.RED;
- F. if( Color.RED.ordinal() < Color.BLUE.ordinal() ) {}

**Answer: BF**

#### Question 25

Given:

```
11. public class Rainbow {  
12. public enum MyColor {  
13. RED(0xff0000), GREEN(0x00ff00), BLUE(0x0000ff);  
14. private final int rgb;  
15. MyColor(int rgb) { this.rgb = rgb; }  
16. public int getRGB() { return rgb; }  
17. };
```

18. `public static void main(String[] args) {`

19. `// insert code here`

20. `}`

21. `}`

Which code fragment, inserted at line 19, allows the Rainbow class to compile?

A. `MyColor skyColor = BLUE;`

B. `MyColor treeColor = MyColor.GREEN;`

C. `if(RED.getRGB() < BLUE.getRGB()) { }`

D. Compilation fails due to other error(s) in the code.

E. `MyColor purple = new MyColor(0xff00ff);`

F. `MyColor purple = MyColor.BLUE + MyColor.RED;`

**Answer:** B

#### Question 26

Given:

21. `class Money {`

22. `private String country = "Canada";`

23. `public String getC() { return country; }`

24. `}`

25. `class Yen extends Money {`

26. `public String getC() { return super.country; }`

27. `}`

28. `public class Euro extends Money {`

29. `public String getC(int x) { return super.getC(); }`

30. `public static void main(String[] args) {`

31. `System.out.print(new Yen().getC() + " " + new Euro().getC());`

32. `}`



33. }

**What is the result?**

- A. Canada
- B. null Canada
- C. Canada null
- D. Canada Canada
- E. Compilation fails due to an error on line 26.
- F. Compilation fails due to an error on line 29.

**Answer:** E

## 3. Operators

Q: 01 Given:

```
11. public class Test {  
12.     public static void main(String [] args) {  
13.         int x = 5;  
14.         boolean b1 = true;  
15.         boolean b2 = false;  
16.  
17.         if ((x == 4) && !b2 )  
18.             System.out.print("1 ");  
19.         System.out.print("2 ");  
20.         if ((b2 = true) && b1 )  
21.             System.out.print("3 ");  
22.     }  
23. }
```

What is the result?

- A. 2
- B. 3
- C. 1 2
- D. 2 3
- E. 1 2 3
- F. Compilation fails.
- G. An exception is thrown at runtime.

**Answer: D**

**Q: 02** Given the command line java Pass2 and:

```
15. public class Pass2 {  
16. public void main(String [] args) {  
17. int x = 6;  
18. Pass2 p = new Pass2();  
19. p.doStuff(x);  
20. System.out.print(" main x = " + x);  
21. }  
22.  
23. void doStuff(int x) {  
24. System.out.print(" doStuff x = " + x++);  
25. }  
26. }
```

**What is the result?**

- A. Compilation fails.
- B. An exception is thrown at runtime.
- C. doStuff x = 6 main x = 6
- D. doStuff x = 6 main x = 7
- E. doStuff x = 7 main x = 6
- F. doStuff x = 7 main x = 7

**Answer: B**

**Q: 03** Given:

```
13. public class Pass {
```

14. `public static void main(String [] args) {`

15. `int x = 5;`

16. `Pass p = new Pass();`

17. `p.doStuff(x);`

18. `System.out.print(" main x = " + x);`

19. `}`

20.

21. `void doStuff(int x) {`

22. `System.out.print(" doStuff x = " + x++);`

23. `}`

24. `}`

**What is the result?**

A. Compilation fails.

B. An exception is thrown at runtime.

C. doStuff x = 6 main x = 6

D. doStuff x = 5 main x = 5

E. doStuff x = 5 main x = 6

F. doStuff x = 6 main x = 5

**Answer: D**

**Question: 04**

**Given:**

42. `public class ClassA {`

43. `public int getValue() {`

44. `int value=0;`

45. `boolean setting = true;`

46. `String title="Hello";`

47. if (value || (setting && title == "Hello")) { return 1; }

48. if (value == 1 & title.equals("Hello")) { return 2; }

49. }

50. }

And:

70. ClassA a = new ClassA();

71. a.getValue();

What is the result?

A. 1

B. 2

C. Compilation fails.

D. The code runs with no output.

E. An exception is thrown at runtime.

Answer: C

Question: 05

Given:

3. public class Batman {

4. int squares = 81;

5. public static void main(String[] args) {

6. new Batman().go();

7. }

8. void go() {

9. incr(++squares);

10. System.out.println(squares);

11. }

12. void incr(int squares) { squares += 10; }

13. }

What is the result?

- A. 81
- B. 82
- C. 91
- D. 92
- E. Compilation fails.
- F. An exception is thrown at runtime.

**Answer:** B

**Question: 06**

**Given:**

1. class ClassA {
2. public int numberOfInstances;
3. protected ClassA(int numberOfInstances) {
4. this.numberOfInstances = numberOfInstances;
5. }
6. }
7. public class ExtendedA extends ClassA {
8. private ExtendedA(int numberOfInstances) {
9. super(numberOfInstances);
10. }
11. public static void main(String[] args) {
12. ExtendedA ext = new ExtendedA(420);
13. System.out.print(ext.numberOfInstances);
14. }
15. }

**Which statement is true?**

- A. 420 is the output.
- B. An exception is thrown at runtime.
- C. All constructors must be declared public.
- D. Constructors CANNOT use the private modifier.
- E. Constructors CANNOT use the protected modifier.

**Answer: A**

**Question: 07**

**Given:**

- 3. `public class Spock {`
- 4. `public static void main(String[] args) {`
- 5. `Long tail = 2000L;`
- 6. `Long distance = 1999L;`
- 7. `Long story = 1000L;`
- 8. `if((tail > distance) ^ ((story * 2) == tail))`
- 9. `System.out.print("1");`
- 10. `if(((distance + 1 != tail) ^ ((story * 2) == distance))`
- 11. `System.out.print("2");`
- 12. `}`
- 13. `}`

**What is the result?**

- A. 1
- B. 2
- C. 12
- D. Compilation fails.
- E. No output is produced.
- F. An exception is thrown at runtime.

**Answer:** E



## 4. Flow Control

Q: 01 Given:

```
10. public class Bar {  
11. static void foo( int... x ) {  
12. // insert code here  
13. }  
14. }
```

Which two code fragments, inserted independently at line 12, will allow the class to compile?  
(Choose

two.)

- A. foreach( x ) System.out.println(z);
- B. for( int z : x ) System.out.println(z);
- C. while( x.hasNext() ) System.out.println( x.next() );
- D. for( int i=0; i< x.length; i++ ) System.out.println(x[i]);

Answer: B, D

Q: 02 Click the Task button.

Place the correct Code in the Code Sample to achieve the expected results.

### Expected Results

Output: 1 2 4 8 16 32

### Code Sample

```
int [] y = { 1, 2, 4, 8, 16, 32 };
System.out.print('Output: ');
```

Place here

```
System.out.print(x);
System.out.print(' ');
}
```

### Code

```
for(int x : y) { } for(int x = y[0]) { }
foreach (y as x) { } foreach (int x : y) { }
for(int x=1; x=y[0]; x++) { }
```

### Solution:

```
int [] y={1,2,4,8,16,32};
```

```
System.out.print("output : ");
```

```
for(int x : y) {
```

```
System.out.println(x);
```

```
System.out.println(" ");
```

Q: 03 Given:

25. int x = 12;

26. while (x < 10) {

27. x--;

28. }

29. System.out.print(x);

**What is the result?**

- A. 0
- B. 10
- C. 12
- D. Line 29 will never be reached.

**Answer: C**

**Q: 04 Given:**

```
11. public static void main(String[] args) {  
12.     Object obj = new int[] { 1, 2, 3 };  
13.     int[] someArray = (int[])obj;  
14.     for (int i : someArray) System.out.print(i + " ");  
15. }
```

**What is the result?**

- A. 1 2 3
- B. Compilation fails because of an error in line 12.
- C. Compilation fails because of an error in line 13.
- D. Compilation fails because of an error in line 14.
- E. A ClassCastException is thrown at runtime.

**Answer: A**

**Q: 05 Given:**

```
11. public static void main(String[] args) {  
12.     for (int i = 0; i <= 10; i++) {  
13.         if (i > 6) break;
```

14. }

15. `System.out.println(i);`

16. }

**What is the result?**

A. 6

B. 7

C. 10

D. 11

E. Compilation fails.

F. An exception is thrown at runtime.

**Answer: E**

**Q: 06 Given:**

11. `public static void main(String[] args) {`

12. `Integer i = new Integer(1) + new Integer(2);`

13. `switch(i) {`

14. `case 3: System.out.println("three"); break;`

15. `default: System.out.println("other"); break;`

16. }

17. }

**What is the result?**

A. three

B. other

C. An exception is thrown at runtime.

D. Compilation fails because of an error on line 12.

E. Compilation fails because of an error on line 13.

F. Compilation fails because of an error on line 15.

**Answer: A**

**Q: 07 Given:**

```
10. public class ClassA {  
11.     public void count(int i) {  
12.         count(++i);  
13.     }  
14. }
```

**And:**

```
20. ClassA a = new ClassA();  
21. a.count(3);
```

**Which exception or error should be thrown by the virtual machine?**

- A. StackOverflowError
- B. NullPointerException
- C. NumberFormatException
- D. IllegalArgumentException
- E. ExceptionInInitializerError

**Answer: A**

**Q: 08 Given:**

```
22. public void go() {  
23.     String o = "";  
24.     z:  
25.     for(int x = 0; x < 3; x++) {  
26.         for(int y = 0; y < 2; y++) {  
27.             if(x==1) break;  
28.             if(x==2 && y==1) break z;  
29.             o = o + x + y;
```

30. }

31. }

32. `System.out.println(o);`

33. }

What is the result when the `go()` method is invoked?

A. 00

B. 0001

C. 000120

D. 00012021

E. Compilation fails.

F. An exception is thrown at runtime.

**Answer: C**

**Q: 09 Given:**

3. `public class Breaker {`

4. `static String o = "";`

5. `public static void main(String[] args) {`

6. `z:`

7. `o = o + 2;`

8. `for(int x = 3; x < 8; x++) {`

9. `if(x==4) break;`

10. `if(x==6) break z;`

11. `o = o + x;`

12. `}`

13. `System.out.println(o);`

14. `}`

15. }

What is the result?

- A. 23
- B. 234
- C. 235
- D. 2345
- E. 2357
- F. 23457
- G. Compilation fails.

**Answer:** G

**Q: 10 Given:**

35. `int x = 10;`

36. `do { 37. x--;`

38. `} while (x < 10);`

How many times will line 37 be executed?

- A. ten times
- B. zero times
- C. one to nine times
- D. more than ten times

**Answer:** D

## 5. Exception Handling

Q: 01 Given:

```
11. public static void parse(String str) {  
12. try {  
13. float f = Float.parseFloat(str);  
14. } catch (NumberFormatException nfe) {  
15. f = 0;  
16. } finally {  
17. System.out.println(f);  
18. }  
19. }  
20. public static void main(String[] args) {  
21. parse("invalid");  
22. }
```

What is the result?

- A. 0.0
- B. Compilation fails.
- C. A ParseException is thrown by the parse method at runtime.
- D. A NumberFormatException is thrown by the parse method at runtime.

Answer: B

Q: 02 Given:

```
84. try {  
85. ResourceConnection con = resourceFactory.getConnection();  
86. Results r = con.query("GET INFO FROM CUSTOMER");
```



```
87. info = r.getData();
88. con.close();
89. } catch (ResourceException re) {
90. errorLog.write(re.getMessage());
91. }
92. return info;
```

**Which statement is true if a ResourceException is thrown on line 86?**

- A. Line 92 will not execute.
- B. The connection will not be retrieved in line 85.
- C. The resource connection will not be closed on line 88.
- D. The enclosing method will throw an exception to its caller.

**Answer: C**

**Q: 03 Given:**

```
31. // some code here
32. try {
33. // some code here
34. } catch (SomeException se) {
35. // some code here
36. } finally {
37. // some code here
38. }
```

**Under which three circumstances will the code on line 37 be executed? (Choose three.)**

- A. The instance gets garbage collected.
- B. The code on line 33 throws an exception.

- C. The code on line 35 throws an exception.
- D. The code on line 31 throws an exception.
- E. The code on line 33 executes successfully.

**Answer: B, C, E**

**Q: 04 Given:**

```
11. class A {  
12.     public void process() { System.out.print("A,"); }  
13. class B extends A {  
14.     public void process() throws IOException {  
15.         super.process();  
16.         System.out.print("B,");  
17.         throw new IOException();  
18.     }  
19.     public static void main(String[] args) {  
20.         try { new B().process(); }  
21.         catch (IOException e) { System.out.println("Exception"); }}
```

**What is the result?**

- A. Exception
- B. A,B,Exception
- C. Compilation fails because of an error in line 20.
- D. Compilation fails because of an error in line 14.
- E. A NullPointerException is thrown at runtime.

**Answer: D**

**Q: 05 Given:**

```
11. static void test() throws Error {
```

```
12. if (true) throw new AssertionError();
13. System.out.print("test ");
14. }
15. public static void main(String[] args) {
16. try { test(); }
17. catch (Exception ex) { System.out.print("exception "); }
18. System.out.print("end ");
19. }
```

**What is the result?**

- A. end
- B. Compilation fails.
- C. exception end
- D. exception test end
- E. A Throwable is thrown by main.
- F. An Exception is thrown by main.

**Answer: E**

**Q: 06 Given:**

```
11. Float pi = new Float(3.14f);
12. if (pi > 3) {
13. System.out.print("pi is bigger than 3. ");
14. }
15. else {
16. System.out.print("pi is not bigger than 3. ");
17. }
18. finally {
19. System.out.println("Have a nice day.");
```

20. }

**What is the result?**

- A. Compilation fails.
- B. pi is bigger than 3.
- C. An exception occurs at runtime.
- D. pi is bigger than 3. Have a nice day.
- E. pi is not bigger than 3. Have a nice day.

**Answer: A**

**Q: 07 Given:**

```
11. public static void main(String[] args) {  
12. try {  
13. args = null;  
14. args[0] = "test";  
15. System.out.println(args[0]);  
16. } catch (Exception ex) {  
17. System.out.println("Exception");  
18. } catch (NullPointerException npe) {  
19. System.out.println("NullPointerException");  
20. }  
21. }
```

**What is the result?**

- A. test
- B. Exception
- C. Compilation fails.

D. NullPointerException

**Answer: C**

**Q:08 Click the Exhibit button.**

**Given:**

```
25. try {  
26. A a = new A();  
27. a.method1();  
28. } catch (Exception e) {  
29. System.out.print("an error occurred");  
30. }
```

**Which two statements are true if a NullPointerException is thrown on line 3 of class C? (Choose two.)**

```
1. public class A {  
2.     public void method1() {  
3.         B b = new B();  
4.         b.method2();  
5.         // more code here  
6.     }  
7. }  
  
1. public class B {  
2.     public void method2() {  
3.         C c = new C();  
4.         c.method3();  
5.         // more code here  
6.     }  
7. }  
  
1. public class C {  
2.     public void method3() {  
3.         // more code here  
4.     }  
5. }
```

A. The application will crash.

B. The code on line 29 will be executed.

- C. The code on line 5 of class A will execute.
- D. The code on line 5 of class B will execute.
- E. The exception will be propagated back to line 27.

**Answer: B, E**

**Q:09 Given:**

```
11. static void test() throws RuntimeException {  
12. try {  
13. System.out.print("test ");  
14. throw new RuntimeException();  
15. }  
16. catch (Exception ex) { System.out.print("exception "); }  
17. }  
18. public static void main(String[] args) {  
19. try { test(); }  
20. catch (RuntimeException ex) { System.out.print("runtime "); }  
21. System.out.print("end ");  
22. }
```

**What is the result?**

- A. test end
- B. Compilation fails.
- C. test runtime end
- D. test exception end
- E. A Throwable is thrown by main at runtime.

**Answer: D**

**Q:10 Given:**

```
33. try {  
34. // some code here  
35. } catch (NullPointerException e1) {  
36. System.out.print("a");  
37. } catch (RuntimeException e2) {  
38. System.out.print("b");  
39. } finally {  
40. System.out.print("c");  
41. }
```

What is the result if a NullPointerException occurs on line 34?

- A. c
- B. a
- C. ab
- D. ac
- E. bc
- F. abc

**Answer: D**

**Q:11 Given:**

```
10. public class Foo {  
11. static int[] a;  
12. static { a[0]=2; }  
13. public static void main( String[] args ) {}  
14. }
```

Which exception or error will be thrown when a programmer attempts to run this code?

- A. java.lang.StackOverflowError
- B. java.lang.IllegalStateException
- C. java.lang.ExceptionInInitializerError
- D. java.lang.ArrayIndexOutOfBoundsException

**Answer: C**

**Q: 12 Given:**

```
11. static void test() {  
12. try {  
13. String x = null;  
14. System.out.print(x.toString() + " ");  
15. }  
16. finally { System.out.print("finally "); }  
17. }  
18. public static void main(String[] args) {  
19. try { test(); }  
20. catch (Exception ex) { System.out.print("exception "); }  
21. }
```

**What is the result?**

- A. null
- B. finally
- C. null finally
- D. Compilation fails.
- E. finally exception

**Answer: E**



Q: 13 Click the Exhibit button.

Given:

31. public void method() {

32. A a = new A();

33. a.method1();

34. }

Which statement is true if a TestException is thrown on line 3 of class B?

```
1. public class A {
2.     public void method1() {
3.         try {
4.             B b = new B();
5.             b.method2();
6.             // more code here
7.         } catch (TestException te) {
8.             throw new RuntimeException(te);
9.         }
10.    }
11. }

1. public class B {
2.     public void method2() throws
TestException {
3.         // more code here
4.     }
5. }

1. public class TestException extends
Exception {
2. }
```

- A. Line 33 must be called within a try block.
- B. The exception thrown by method1 in class A is not required to be caught.
- C. The method declared on line 31 must be declared to throw a RuntimeException.
- D. On line 5 of class A, the call to method2 of class B does not need to be placed in a try/catch block.

**Answer: B**

**Q: 14 Click the Exhibit button.**

**Which statement is true about the two classes?**

SomeException:

```
1. public class SomeException {  
2. }
```

Class A:

```
1. public class A {  
2.     public void doSomething() { }  
3. }
```

Class B:

```
1. public class B extends A {  
2.     public void doSomething() throws  
   SomeException { }  
3. }
```

- A. Compilation of both classes will fail.
- B. Compilation of both classes will succeed.
- C. Compilation of class A will fail. Compilation of class B will succeed.
- D. Compilation of class B will fail. Compilation of class A will succeed.

**Answer: D**

**Question: 15**

**Click the Exhibit button.**

**Class TestException**

```
1. public class TestException extends Exception {  
2. }
```

**Class A:**

```
1. public class A {  
2.  
3. public String sayHello(String name) throws TestException {
```

4.

5. if(name == null) {

6. throw new TestException();

7. }

8.

9. return "Hello "+ name;

10. }

11.

12. }

A programmer wants to use this code in an application:

45. A a=new A();

46. System.out.println(a.sayHello("John"));

Which two are true? (Choose two.)

A. Class A will not compile.

B. Line 46 can throw the unchecked exception TestException.

C. Line 45 can throw the unchecked exception TestException.

D. Line 46 will compile if the enclosing method throws a TestException.

E. Line 46 will compile if enclosed in a try block, where TestException is caught.

**Answer: DE**

**Question:16**

**Given:**

11.class A {

12. public void process() { System.out.print("A "); } }

13. class B extends A {

14. public void process() throws RuntimeException {

15. super.process();

16. if (true) throw new RuntimeException();

```
17. System.out.print("B"); }}  
18. public static void main(String[] args) {  
19. try { ((A)new B()).process(); }  
20. catch (Exception e) { System.out.print("Exception "); }  
21. }
```

**What is the result?**

- A. Exception
- B. A Exception
- C. A Exception B
- D. A B Exception
- E. Compilation fails because of an error in line 14.
- F. Compilation fails because of an error in line 19.

**Answer: B**

**Question:17**

**which two code fragments are most likely to cause a StackOverflowError? (Choose two.)**

- A. 

```
int []x = {1,2,3,4,5};  
for(int y = 0; y < 6; y++)  
System.out.println(x[y]);
```
- B. 

```
static int[] x = {7,6,5,4};  
static { x[1] = 8;  
x[4] = 3; }
```
- C. 

```
for(int y = 10; y < 10; y++)  
doStuff(y);
```
- D. 

```
void doOne(int x) { doTwo(x); }  
void doTwo(int y) { doThree(y); }  
void doThree(int z) { doTwo(z); }
```

E. for(int x = 0; x < 1000000000; x++)

doStuff(x);

F. void counter(int i) { counter(++i); }

**Answer:** D,F

**Question:18**

**which can appropriately be thrown by a programmer using Java SE technology to create a desktop application?**

A. ClassCastException

B. NullPointerException

C. NoClassDefFoundError

D. NumberFormatException

E. ArrayIndexOutOfBoundsException

**Answer: D**

## 6. Asserions

Q: 01 Given:

```
8. public class test {  
9.     public static void main(String [] a) {  
10.         assert a.length == 1;  
11.     }  
12. }
```

Which two will produce an AssertionError? (Choose two.)

- A. java test
- B. java -ea test
- C. java test file1
- D. java -ea test file1
- E. java -ea test file1 file2
- F. java -ea:test test file1

Answer: B, E

Q: 02 Given a method that must ensure that its parameter is not null:

```
11. public void someMethod(Object value) {  
12.     // check for null value  
...  
20. System.out.println(value.getClass());  
21. }
```

What, inserted at line 12, is the appropriate way to handle a null value?

- A. `assert value == null;`
- B. `assert value != null, "value is null";`
- C. `if (value == null) {  
throw new AssertionError("value is null");  
}`
- D. `if (value == null) {  
throw new IllegalArgumentException("value is null");  
}`

**Answer: D**

**Q: 03 Given:**

- 23. `int z = 5;`
- 24.
- 25. `public void stuff1(int x) {`
- 26. `assert (x > 0);`
- 27. `switch(x) {`
- 28. `case 2: x = 3;`
- 29. `default: assert false; } }`
- 30.
- 31. `private void stuff2(int y) { assert (y < 0); }`
- 32.
- 33. `private void stuff3() { assert (stuff4()); }`
- 34.
- 35. `private boolean stuff4() { z = 6; return false; }`

**Which statement is true?**

- A. All of the assert statements are used appropriately.

- B. Only the assert statement on line 31 is used appropriately.
- C. The assert statements on lines 29 and 31 are used appropriately.
- D. The assert statements on lines 26 and 29 are used appropriately.
- E. The assert statements on lines 29 and 33 are used appropriately.
- F. The assert statements on lines 29, 31, and 33 are used appropriately.
- G. The assert statements on lines 26, 29, and 31 are used appropriately.

**Answer: C**

**Question: 04**

Click the Exhibit button.

```
1. public class Test {  
2.  
3. public static void main(String [] args) {  
4. boolean assert = true;  
5. if(assert) {  
6. System.out.println("assert is true");  
7. }  
8. }  
9.  
10. }
```

**Given:**

```
javac -source 1.3 Test.java
```

**What is the result?**

- A. Compilation fails.
- B. Compilation succeeds with errors.
- C. Compilation succeeds with warnings.



D. Compilation succeeds without warnings or errors.

**Answer: C**

**Question: 05**

**Given:**

1. `public class Mule {`
2. `public static void main(String[] args) {`
3. `boolean assert = true;`
4. `if(assert) {`
5. `System.out.println("assert is true");`
6. `}`
7. `}`
8. `}`

**Which command-line invocations will compile?**

- A. `javac Mule.java`
- B. `javac -source 1.3 Mule.java`
- C. `javac -source 1.4 Mule.java`
- D. `javac -source 1.5 Mule.java`

**Answer: B**

**Question: 06**

**Given:**

1. `public class Donkey2 {`
2. `public static void main(String[] args) {`
3. `boolean assertsOn = true;`
4. `assert (assertsOn) : assertsOn = true;`
5. `if(assertsOn) {`

6. `System.out.println("assert is on");`

7. `}`

8. `}`

9. `}`

If class Donkey is invoked twice, the first time without assertions enabled, and the second time with assertions enabled, what are the results?

A. no output

B. no output

assert is on

C. assert is on

D. no output

An AssertionError is thrown.

E. assert is on

An AssertionError is thrown.

**Answer: C**

**Question: 07**

**Given:**

1. `public class Donkey {`

2. `public static void main(String[] args) {`

3. `boolean assertsOn = false;`

4. `assert (assertsOn) : assertsOn = true;`

5. `if(assertsOn) {`

6. `System.out.println("assert is on");`

7. `}`

8. `}`

9. `}`

If class Donkey is invoked twice, the first time without assertions enabled, and the second

time with assertions enabled, what are the results?

A. no output

B. no output

assert is on

C. assert is on

D. no output

An AssertionError is thrown.

E. assert is on

An AssertionError is thrown.

**Answer:** D

**Question: 08**

**Given:**

11. `public void go(int x) {`

12. `assert (x > 0);`

13. `switch(x) {`

14. `case 2: ;`

15. `default: assert false;`

16. `}`

17. `}`

**18. private void go2(int x) { assert (x < 0); }**

**Which statement is true?**

- A. All of the assert statements are used appropriately.
- B. Only the assert statement on line 12 is used appropriately.
- C. Only the assert statement on line 15 is used appropriately.
- D. Only the assert statement on line 18 is used appropriately.
- E. Only the assert statements on lines 12 and 15 are used appropriately.
- F. Only the assert statements on lines 12 and 18 are used appropriately.
- G. Only the assert statements on lines 15 and 18 are used appropriately.

**Answer: G**

## 7. Garbage Collections

Q: 01 Given:

1. `public class GC {`
2. `private Object o;`
3. `private void doSomethingElse(Object obj) { o = obj; }`
4. `public void doSomething() {`
5. `Object o = new Object();`
6. `doSomethingElse(o);`
7. `o = new Object();`
8. `doSomethingElse(null);`
9. `o = null;`
10. `}`
11. `}`

When the `doSomething` method is called, after which line does the Object created in line 5 become

available for garbage collection?

- A. Line 5
- B. Line 6
- C. Line 7
- D. Line 8
- E. Line 9
- F. Line 10

Answer: D

Q: 02 Given:

```
11. public void genNumbers() {  
12. ArrayList numbers = new ArrayList();  
13. for (int i=0; i<10; i++) {  
14. int value = i * ((int) Math.random());  
15. Integer intObj = new Integer(value);  
16. numbers.add(intObj);  
17. }  
18. System.out.println(numbers);  
19. }
```

Which line of code marks the earliest point that an object referenced by intObj becomes a candidate for

garbage collection?

- A. Line 16
- B. Line 17
- C. Line 18
- D. Line 19
- E. The object is NOT a candidate for garbage collection.

**Answer: D**

Q: 03 Given:

```
11. rbo = new ReallyBigObject();  
12. // more code here
```

13. `rbo = null;`

14. `/* insert code here */`

Which statement should be placed at line 14 to suggest that the virtual machine expend effort toward

recycling the memory used by the object `rbo`?

A. `System.gc();`

B. `Runtime.gc();`

C. `System.freeMemory();`

D. `Runtime.getRuntime().growHeap();`

E. `Runtime.getRuntime().freeMemory();`

**Answer: A**

**Q: 04 Given:**

11. `class Snoochy {`

12. `Boochy booch;`

13. `public Snoochy() { booch = new Boochy(this); }`

14. `}`

15.

16. `class Boochy {`

17. `Snoochy snooch;`

18. `public Boochy(Snoochy s) { snooch = s; }`

19. `}`

**And the statements:**

21. `public static void main(String[] args) {`

22. `Snoochy snoog = new Snoochy();`

23. `snoog = null;`

24. `// more code here`

25. `}`

**Which statement is true about the objects referenced by snoog, snooch, and booch immediately after line 23 executes?**

- A. None of these objects are eligible for garbage collection.
- B. Only the object referenced by booch is eligible for garbage collection.
- C. Only the object referenced by snoog is eligible for garbage collection.
- D. Only the object referenced by snooch is eligible for garbage collection.
- E. The objects referenced by snooch and booch are eligible for garbage collection.

**Answer: E**

**Question: 05**

**Which two are true? (Choose two.)**

- A. A finalizer may NOT be invoked explicitly.
- B. The finalize method declared in class Object takes no action.
- C. `super.finalize()` is called implicitly by any overriding finalize method.
- D. The finalize method for a given object will be called no more than once by the garbage collector.
- E. The order in which finalize will be called on two objects is based on the order in which the two objects became finalizable.

**Answer: BD**

**Question: 06**

**Which statement is true?**

- A. A class's `finalize()` method CANNOT be invoked explicitly.
- B. `super.finalize()` is called implicitly by any overriding `finalize()` method.
- C. The `finalize()` method for a given object is called no more than once by the garbage collector.
- D. The order in which `finalize()` is called on two objects is based on the order in which the two objects became finalizable.



**Answer: C**

**Question: 07**

**Given:**

```
3. interface Animal { void makeNoise(); }  
4. class Horse implements Animal {  
5. Long weight = 1200L;  
6. public void makeNoise() { System.out.println("whinny"); }  
7. }  
8. public class Icelandic extends Horse {  
9. public void makeNoise() { System.out.println("vinny"); }  
10. public static void main(String[] args) {  
11. Icelandic i1 = new Icelandic();  
12. Icelandic i2 = new Icelandic();  
12. Icelandic i3 = new Icelandic();  
13. i3 = i1; i1 = i2; i2 = null; i3 = i1;  
14. }  
15. }
```

**When line 14 is reached, how many objects are eligible for the garbage collector?**

- A. 0
- B. 1
- C. 2
- D. 3
- E. 4
- F. 6

**Answer: E**

Question: 07

Given:

```
5. public class Tahiti {  
6. Tahiti t;  
7. public static void main(String[] args) {  
8. Tahiti t = new Tahiti();  
9. Tahiti t2 = t.go(t);  
10. t2 = null;  
11. // more code here  
12. }  
13. Tahiti go(Tahiti t) {  
14. Tahiti t1 = new Tahiti(); Tahiti t2 = new Tahiti();  
15. t1.t = t2; t2.t = t1; t.t = t2;  
16. return t1;  
17. }  
18. }
```

When line 11 is reached, how many objects are eligible for garbage collection?

- A. 0
- B. 1
- C. 2
- D. 3
- E. Compilation fails.

Answer: A

# 8.OOPs

Q: 01 Click the Task button.

Place the Output Options in the Actual Output Sequence to indicate the output from this code:

```
class Alpha {
    public void foo( String a ) {
        { System.out.print("Alpha:foo "); }
    }
    public void bar( String a ) {
        { System.out.print("Alpha:bar "); }
    }
}

public class Beta extends Alpha {
    public void foo( String a ) {
        { System.out.print("Beta:foo "); }
    }
    public void bar( String a ) {
        { System.out.print("Beta:bar "); }
    }
    public static void main( String[] argv ) {
        Alpha a = new Alpha();
        Beta b = new Beta();
        a.foo( "test" ); b.foo( "test" );
        a.bar( "test" ); b.bar( "test" );
    }
}
```

**Actual Output Sequence**

Place here

Place here

Place here

Place here

**Output Options**

Alpha:foo

Alpha:bar

Beta:foo

Beta:bar

Done

**Solution:**

Alpha.foo      Beta.bar      Beta.foo      Beta.bar

Q: 02 Given:

1. public class Plant {
2. private String name;
3. public Plant(String name) { this.name = name; }
4. public String getName() { return name; }
5. }
1. public class Tree extends Plant {
2. public void growFruit() { }

3. `public void dropLeaves() { }`

4. `}`

**Which statement is true?**

- A. The code will compile without changes.
- B. The code will compile if `public Tree() { Plant(); }` is added to the Tree class.
- C. The code will compile if `public Plant() { Tree(); }` is added to the Plant class.
- D. The code will compile if `public Plant() { this("fern"); }` is added to the Plant class.
- E. The code will compile if `public Plant() { Plant("fern"); }` is added to the Plant class.

**Answer: D**

**Q: 03 Click the Exhibit button.**

**Which statement is true about the classes and interfaces in the exhibit?**

```
1. public interface A {
2.     public void doSomething(String thing);
3. }

1. public class AImpl implements A {
2.     public void doSomething(String msg) { }
3. }

1. public class B {
2.     public A doit() {
3.         // more code here
4.     }
5.
6.     public String execute() {
7.         // more code here
8.     }
9. }

1. public class C extends B {
2.     public AImpl doit() {
3.         // more code here
4.     }
5.
6.     public Object execute() {
7.         // more code here
8.     }
9. }
```

- A. Compilation will succeed for all classes and interfaces.
- B. Compilation of class C will fail because of an error in line 2.
- C. Compilation of class C will fail because of an error in line 6.
- D. Compilation of class Almpl will fail because of an error in line 2.

**Answer: C**

**Q: 04 Given:**

```
11. public class Yikes {  
12.  
13.     public static void go(Long n) {System.out.println("Long ");}  
14.     public static void go(Short n) {System.out.println("Short ");}  
15.     public static void go(int n) {System.out.println("int ");}  
16.     public static void main(String [] args) {  
17.         short y = 6;  
18.         long z = 7;  
19.         go(y);  
20.         go(z);  
21.     }  
22. }
```

**What is the result?**

- A. int Long
- B. Short Long
- C. Compilation fails.
- D. An exception is thrown at runtime.

**Answer: A**

**Q: 05 Click the Task button.**

Given:

```
class A {
    String name = "A";
    String getName() {
        return name;
    }
    String greeting(){
        return "class A";
    }
}
class B extends A {
    String name = "B";
    String greeting() {
        return "class B";
    }
}
public class Client {
    public static void main( String[] args ) {
        A a = new A();
        B b = new B();
        System.out.println(a.greeting() + " has name " + a.getName());
        System.out.println(b.greeting() + " has name " + b.getName());
    }
}
```

Place the names "A" and "B" in the following output.

class  has name

class  has name

**Names**

**Solution:**

class A has name A

class B hasname A

**Q: 06 Which two statements are true about has-a and is-a relationships? (Choose two.)**

- A. Inheritance represents an is-a relationship.
- B. Inheritance represents a has-a relationship.
- C. Interfaces must be used when creating a has-a relationship.
- D. Instance variables can be used when creating a has-a relationship.

**Answer: A, D**

**Q: 07 Given:**

```
10: public class Hello {  
11: String title;  
12: int value;  
13: public Hello() {  
14: title += " World";  
15: }  
16: public Hello(int value) {  
17: this.value = value;  
18: title = "Hello";  
19: Hello();  
20: }  
21: }
```

**and:**

```
30: Hello c = new Hello(5);  
31: System.out.println(c.title);
```

**What is the result?**

- A. Hello
- B. Hello World
- C. Compilation fails.
- D. Hello World 5
- E. The code runs with no output.
- F. An exception is thrown at runtime.

**Answer: C**

**Q: 08 Given:**

```
1. class Super {
```

2. `private int a;`

3. `protected Super(int a) { this.a = a; }`

4. `}`

...

11. `class Sub extends Super {`

12. `public Sub(int a) { super(a); }`

13. `public Sub() { this.a = 5; }`

14. `}`

Which two, independently, will allow Sub to compile? (Choose two.)

A. Change line 2 to:

`public int a;`

B. Change line 2 to `:protected int a;`

C. Change line 13 to `:public Sub() { this(5); }`

D. Change line 13 to `:public Sub() { super(5); }`

E. Change line 13 to `:public Sub() { super(a); }`

**Answer: C, D**

**Q: 09 Click the Task button.**

Place the Relations on their corresponding Implementation Structures.  
Note: Not all Implementation Structures will be used.

Implementation Structures		Relations
<code>class A {   List&lt;B&gt; b; }</code>	<code>class A extends B,C { }</code>	Car is a Vehicle and Car is a Collectable
<code>class A { }</code>	<code>class A {   B b; C c; }</code>	Car has a SteeringWheel
<code>class A {   B b; }</code>	<code>class A implements B,C { }</code>	Car has Wheels
<code>class A extends B { }</code>		Mini is a Car
<div>Core</div>		Car is an Object





```
7. public class ExtendedA extends ClassA {  
8.     private ExtendedA(int numberOfInstances) {  
9.         super(numberOfInstances);  
10.    }  
11.    public static void main(String[] args) {  
12.        ExtendedA ext = new ExtendedA(420);  
13.        System.out.print(ext.numberOfInstances);  
14.    }  
15. }
```

**Which statement is true?**

- A. 420 is the output.
- B. An exception is thrown at runtime.
- C. All constructors must be declared public.
- D. Constructors CANNOT use the private modifier.
- E. Constructors CANNOT use the protected modifier.

**Answer: A**

**Q: 11 Given:**

```
1. interface A { public void aMethod(); }  
2. interface B { public void bMethod(); }  
3. interface C extends A,B { public void cMethod(); }  
4. class D implements B {  
5.     public void bMethod(){  
6.     }  
7. class E extends D implements C {  
8.     public void aMethod(){
```

```
9. public void bMethod(){  
10. public void cMethod(){  
11. }
```

What is the result?

- A. Compilation fails because of an error in line 3.
- B. Compilation fails because of an error in line 7.
- C. Compilation fails because of an error in line 9.
- D. If you define `D e = new E()`, then `e.bMethod()` invokes the version of `bMethod()` defined in Line 5.
- E. If you define `D e = (D)(new E())`, then `e.bMethod()` invokes the version of `bMethod()` defined in Line 5.
- F. If you define `D e = (D)(new E())`, then `e.bMethod()` invokes the version of `bMethod()` defined in Line 9.

Answer: F

Q: 12 Given:

```
1. public class Base {  
2. public static final String FOO = "foo";  
3. public static void main(String[] args) {  
4. Base b = new Base();  
5. Sub s = new Sub();  
6. System.out.print(Base.FOO);  
7. System.out.print(Sub.FOO);  
8. System.out.print(b.FOO);  
9. System.out.print(s.FOO);
```

10. `System.out.print(((Base)s).FOO);`

11. `}}`

12. `class Sub extends Base {public static final String FOO="bar";}`

**What is the result?**

A. fofoofoofoofoo

B. foobarfoobarbar

C. foobarfoofoofoo

D. foobarfoobarfoo

E. barbarbarbarbar

F. fofoofooobarbar

G. fofoofooobarfoo

**Answer: D**

**Q: 13 Which two statements are true? (Choose two.)**

A. An encapsulated, public class promotes re-use.

B. Classes that share the same interface are always tightly encapsulated.

C. An encapsulated class allows subclasses to overload methods, but does NOT allow overriding methods.

D. An encapsulated class allows a programmer to change an implementation without affecting outside code.

**Answer: A, D**

**Q: 14 Given:**

11. `public static void test(String str) {`

12. `if (str == null | str.length() == 0) {`

```
13. System.out.println("String is empty");  
14. } else {  
15. System.out.println("String is not empty");  
16. }  
17. }
```

And the invocation:

```
31. test(null);
```

What is the result?

- A. An exception is thrown at runtime.
- B. "String is empty" is printed to output.
- C. Compilation fails because of an error in line 12.
- D. "String is not empty" is printed to output.

Answer: A

Q: 15 Given:

```
12. public class Wow {  
13. public static void go(short n) {System.out.println("short");}  
14. public static void go(Short n) {System.out.println("SHORT");}  
15. public static void go(Long n) {System.out.println(" LONG");}  
16. public static void main(String [] args) {  
17. Short y = 6;  
18. int z = 7;  
19. go(y);  
20. go(z);  
21. }  
22. }
```

**What is the result?**

- A. short LONG
- B. SHORT LONG
- C. Compilation fails.
- D. An exception is thrown at runtime.

**Answer: C**

**Q: 16 Click the Exhibit button.**

**Which three code fragments, added individually at line 29, produce the output 100? (Choose three.)**

```
10. class Inner {
11.     private int x;
12.     public void setX( int x ) { this.x = x; }
13.     public int getX() { return x; }
14. }
15.
16. class Outer {
17.     private Inner y;
18.     public void setY( Inner y ) { this.y = y; }
19.     public Inner getY() { return y; }
20. }
21.
22. public class Gamma {
23.     public static void main( String[] args
24. ) {
25.         Outer o = new Outer();
26.         Inner i = new Inner();
27.         int n = 10;
28.         i.setX( n );
29.         o.setY( i );
30.         // insert code here
31.         System.out.println( o.getY().getX() );
32.     }
```

- A. n = 100;
- B. i.setX( 100 );
- C. o.getY().setX( 100 );

D. i = new Inner(); i.setX( 100 );

E. o.setY( i ); i = new Inner(); i.setX( 100 );

F. i = new Inner(); i.setX( 100 ); o.setY( i );

**Answer: B, C, F**

**Q: 17 Given:**

10. class One {

11. public One() { System.out.print(1); }

12. }

13. class Two extends One {

14. public Two() { System.out.print(2); }

15. }

16. class Three extends Two {

17. public Three() { System.out.print(3); }

18. }

19. public class Numbers{

20. public static void main( String[] argv ) { new Three(); }

21. }

**What is the result when this code is executed?**

A. 1

B. 3

C. 123

D. 321

E. The code runs with no output.

**Answer: C**

**Q: 18** Click the Exhibit button.

**What is the outcome of the code?**

```
1. public class Item {
2.     private String desc;
3.     public String getDescription() { return
desc; }
4.     public void setDescription(String d) {
desc = d; }
5.
6.     public static void modifyDesc(Item
item, String desc) {
7.         item = new Item();
8.         item.setDescription(desc);
9.     }
10.    public static void main(String[] args)
{
11.        Item it = new Item();
12.        it.setDescription("Gobstopper");
13.        Item it2 = new Item();
14.        it2.setDescription("Fizzylifting");
15.        modifyDesc(it,
"Scrumdiddlyumptious");
16.
System.out.println(it.getDescription());
17.
System.out.println(it2.getDescription());
18.    }
19. }
```

- A. Compilation fails.
- B. Gobstopper  
Fizzylifting
- C. Gobstopper  
Scrumdiddlyumptious
- D. Scrumdiddlyumptious  
Fizzylifting
- E. Scrumdiddlyumptious

**Answer: B**

**Q: 19** Given:



```
10. class One {  
11.     public One foo() { return this; }  
12. }  
13. class Two extends One {  
14.     public One foo() { return this; }  
15. }  
16. class Three extends Two {  
17.     // insert method here  
18. }
```

**Which two methods, inserted individually, correctly complete the Three class? (Choose two.)**

- A. public void foo() {}
- B. public int foo() { return 3; }
- C. public Two foo() { return this; }
- D. public One foo() { return this; }
- E. public Object foo() { return this; }

**Answer: C, D**

**Q: 20 Click the Exhibit button.**

**What is the output of the program shown in the exhibit?**

```

10. class Foo {
11.     private int x;
12.     public Foo( int x ) { this.x = x; }
13.     public void setX( int x ) { this.x = x; }
14.     public int getX() { return x; }
15. }
16.
17. public class Gamma {
18.
19.     static Foo fooBar( Foo foo ) {
20.         foo = new Foo( 100 );
21.         return foo;
22.     }
23.
24.     public static void main( String[] args
25. ) {
26.         Foo foo = new Foo( 300 );
27.         System.out.print( foo.getX() + "-" );
28.
29.         Foo fooFoo = fooBar( foo );
30.         System.out.print( foo.getX() + "-" );
31.         System.out.print( fooFoo.getX() + "-" );
32.
33.         foo = fooBar( fooFoo );
34.         System.out.print( foo.getX() + "-" );
35.         System.out.print( fooFoo.getX() );
36.     }

```

- A. 300-100-100-100-100
- B. 300-300-100-100-100
- C. 300-300-300-100-100
- D. 300-300-300-300-100

**Answer: B**

**Q: 21 Click the Task button.**

Add methods to the Beta class to make it compile correctly.

```

class Alpha {
    public void bar( int... x ) { }
    public void bar( int x ) { }
}

```

```

public class Beta extends Alpha {

```

Place here

Place here

Place here

```

}

```

#### Methods

private void bar( int x ) { }

public void bar( int x ) { }

public int bar( String x ) { return 1; }

public Alpha bar( int x ) { }

public void bar( int x, int y ) { }

public int bar( int x ) { return x; }

**Solution:**

```
public void bar(int x){ }  
public int bar(String x){ return 1; }  
public void bar(int x,int y) { }
```

**Q: 22 Given:**

```
10. interface A { public int getValue(); }  
11. class B implements A {  
12. public int getValue() { return 1; }  
13. }  
14. class C extends B {  
15. // insert code here  
16. }
```

**Which three code fragments, inserted individually at line 15, make use of polymorphism? (Choose three.)**

- A. `public void add(C c) { c.getValue(); }`
- B. `public void add(B b) { b.getValue(); }`
- C. `public void add(A a) { a.getValue(); }`
- D. `public void add(A a, B b) { a.getValue(); }`
- E. `public void add(C c1, C c2) { c1.getValue(); }`

**Answer: B, C, D**

**Q: 23 Click the Exhibit button.**

**What is the result?**

```
11. public class Bootchy {
12.     int bootch;
13.     String snootch;
14.
15.     public Bootchy() {
16.         this("snootchy");
17.         System.out.print("first ");
18.     }
19.
20.     public Bootchy(String snootch) {
21.         this(420, "snootchy");
22.         System.out.print("second ");
23.     }
24.
25.     public Bootchy(int bootch, String
snootch) {
26.         this.bootch = bootch;
27.         this.snootch = snootch;
28.         System.out.print("third ");
29.     }
30.
31.     public static void main(String[] args)
{
32.         Bootchy b = new Bootchy();
33.         System.out.print(b.snootch + " " +
b.bootch);
34.     }
35. }
```

- A. snootchy 420 third second first
- B. snootchy 420 first second third
- C. first second third snootchy 420
- D. third second first snootchy 420
- E. third first second snootchy 420
- F. first second first third snootchy 420

**Answer: D**

**Q: 24 Given:**

**10. abstract class A {**

11. `abstract void a1();`

12. `void a2() { }`

13. `}`

14. `class B extends A {`

15. `void a1() { }`

16. `void a2() { }`

17. `}`

18. `class C extends B { void c1() { } }`

and:

`A x = new B(); C y = new C(); A z = new C();`

What are four valid examples of polymorphic method calls? (Choose four.)

A. `x.a2();`

B. `z.a2();`

C. `z.c1();`

D. `z.a1();`

E. `y.c1();`

F. `x.a1();`

**Answer: A, B, D, F**

**Q: 25 Click the Exhibit button.**

**What is the result?**

```
1. public class SimpleCalc {
2.     public int value;
3.     public void calculate() { value += 7; }
4. }
```

And:

```
1. public class MultiCalc extends
SimpleCalc{
2.     public void calculate() { value -= 3; }
3.     public void calculate(int multiplier) {
4.         calculate();
5.         super.calculate();
6.         value *= multiplier;
7.     }
8.     public static void main(String[] args)
{
9.         MultiCalc calculator = new
MultiCalc();
10.        calculator.calculate(2);
11.        System.out.println("Value is: " +
calculator.value);
12.    }
13. }
```

- A. Value is: 8
- B. Compilation fails.
- C. Value is: 12
- D. Value is: -12
- E. The code runs with no output.
- F. An exception is thrown at runtime.

**Answer: A**

**Q: 26 Given:**

- 20. public class CreditCard {
- 21.
- 22. private String cardID;
- 23. private Integer limit;
- 24. public String ownerName;
- 25.
- 26. public void setCardInformation(String cardID,

```
27. String ownerName,  
28. Integer limit) {  
29. this.cardID = cardID;  
30. this.ownerName = ownerName;  
31. this.limit = limit;  
32. }  
33. }
```

**Which statement is true?**

- A. The class is fully encapsulated.
- B. The code demonstrates polymorphism.
- C. The ownerName variable breaks encapsulation.
- D. The cardID and limit variables break polymorphism.
- E. The setCardInformation method breaks encapsulation.

**Answer: C**

**Q: 27**Given:

```
11. class Animal { public String noise() { return "peep"; } }  
12. class Dog extends Animal {  
13. public String noise() { return "bark"; }  
14. }  
15. class Cat extends Animal {  
16. public String noise() { return "meow"; }  
17. }  
...  
30. Animal animal = new Dog();  
31. Cat cat = (Cat)animal;  
32. System.out.println(cat.noise());
```

**What is the result?**

- A. peep
- B. bark
- C. meow
- D. Compilation fails.
- E. An exception is thrown at runtime.

**Answer: E**

**Q: 28 Click the Task button.**

Place the Types in one of the Type columns, and the Relationships in the Relationship column, to define appropriate has-a and is-a relationships.

Type	Relationship	Type	Relationships	Types
Place here	Place here	Animal	is-a	Dog
Forest	Place here	Place here	has-a	Side
Rectangle	Place here	Place here		Tail
Place here	Place here	Programming Book		Square
				Tree
				Book
				Java Book
				Pen

Done

**Solution:**

Dog            is-a            Animal

Forest   has-a            Tree

Rectangle   has-a            Side

JavaBook   is-a            ProgrammingBook



**Q: 29 Which three statements are true? (Choose three.)**

- A. A final method in class X can be abstract if and only if X is abstract.
- B. A protected method in class X can be overridden by any subclass of X.
- C. A private static method can be called only within other static methods in class X.
- D. A non-static public final method in class X can be overridden in any subclass of X.
- E. A public static method in class X can be called by a subclass of X without explicitly referencing the class X.
- F. A method with the same signature as a private final method in class X can be implemented in a subclass of X.
- G. A protected method in class X can be overridden by a subclass of A only if the subclass is in the same package as X.

**Answer: B, E, F**

**Q: 30 Which four statements are true? (Choose four.)**

- A. Has-a relationships should never be encapsulated.
- B. Has-a relationships should be implemented using inheritance.
- C. Has-a relationships can be implemented using instance variables.
- D. Is-a relationships can be implemented using the extends keyword.
- E. Is-a relationships can be implemented using the implements keyword.
- F. The relationship between Movie and Actress is an example of an is-a relationship.
- G. An array or a collection can be used to implement a one-to-many has-a relationship.

**Answer: C, D, E, G**

**Q: 31 Given:**

**10. interface Jumper { public void jump(); }**

...

**20. class Animal { }**

...

30. class Dog extends Animal {

31. Tail tail;

32. }

...

40. class Beagle extends Dog implements Jumper{

41. public void jump() {} 42. }

...

50. class Cat implements Jumper{

51. public void jump() {}

52. }

**Which three are true? (Choose three.)**

A. Cat is-a Animal

B. Cat is-a Jumper

C. Dog is-a Animal

D. Dog is-a Jumper

E. Cat has-a Animal

F. Beagle has-a Tail

G. Beagle has-a Jumper

**Answer: B, C, F**

**Q: 32 Click the Exhibit button.**

**What two must the programmer do to correct the compilation errors? (Choose two.)**

```
1. public class Car {
2.     private int wheelCount;
3.     private String vin;
4.     public Car(String vin) {
5.         this.vin = vin;
6.         this.wheelCount = 4;
7.     }
8.     public String drive() {
9.         return "zoom-zoom";
10.    }
11.    public String getInfo() {
12.        return "VIN: " + vin + " wheels: " +
wheelCount;
13.    }
14. }
```

And:

```
1. public class MeGo extends Car {
2.     public MeGo(String vin) {
3.         this.wheelCount = 3;
4.     }
5. }
```

- A. insert a call to this() in the Car constructor
- B. insert a call to this() in the MeGo constructor
- C. insert a call to super() in the MeGo constructor
- D. insert a call to super(vin) in the MeGo constructor
- E. change the wheelCount variable in Car to protected
- F. change line 3 in the MeGo class to super.wheelCount = 3;

**Answer: D, E**

**Q: 33 Given:**

```
10. public class SuperCalc {
11.     protected static int multiply(int a, int b) { return a * b;}
12. }
```

and:

```
20. public class SubCalc extends SuperCalc{
21.     public static int multiply(int a, int b) {
22.         int c = super.multiply(a, b);
```

23. return c;

24. }

25. }

and:

30. SubCalc sc = new SubCalc ();

31. System.out.println(sc.multiply(3,4));

32. System.out.println(SubCalc.multiply(2,2));

What is the result?

A. 12

4

B. The code runs with no output.

C. An exception is thrown at runtime.

D. Compilation fails because of an error in line 21.

E. Compilation fails because of an error in line 22.

F. Compilation fails because of an error in line 31.

Answer: E

Q: 34 Given:

1. public class Blip {

2. protected int blipvert(int x) { return 0; }

3. }

4. class Vert extends Blip {

5. // insert code here

6. }

Which five methods, inserted independently at line 5, will compile? (Choose five.)

A. public int blipvert(int x) { return 0; }

- B. private int blipvert(int x) { return 0; }
- C. private int blipvert(long x) { return 0; }
- D. protected long blipvert(int x) { return 0; }
- E. protected int blipvert(long x) { return 0; }
- F. protected long blipvert(long x) { return 0; }
- G. protected long blipvert(int x, int y) { return 0; }

**Answer: A, C, E, F, G**

**Q: 35 Given:**

1. class Pizza {
2. java.util.ArrayList toppings;
3. public final void addTopping(String topping) {
4. toppings.add(topping);
5. }
6. }
7. public class PepperoniPizza extends Pizza {
8. public void addTopping(String topping) {
9. System.out.println("Cannot add Toppings");
10. }
11. public static void main(String[] args) {
12. Pizza pizza = new PepperoniPizza();
13. pizza.addTopping("Mushrooms");
14. }
15. }

**What is the result?**

Given:

```
10. public class Pizza {
11.     ArrayList toppings;
12.
13.     public final void addTopping(String
    topping) {
14.         toppings.add(topping);
15.     }
16.
17.     public void removeTopping(String
    topping) {
18.         toppings.remove(topping);
19.     }
20. }
```

And:

```
30. class PepperoniPizza extends Pizza {
31.     public void addTopping(String topping) {
32.         System.out.println("Cannot add
    Toppings");
33.     }
34.
35.     public void removeTopping(String
    topping) {
36.         System.out.println("Cannot remove
    Pepperoni");
37.     }
38. }
```

And:

```
50. Pizza pizza = new PepperoniPizza();
51. pizza.addTopping("Mushrooms");
52. pizza.removeTopping("Pepperoni");
```

- A. Compilation fails.
- B. Cannot add Toppings
- C. The code runs with no output.
- D. A NullPointerException is thrown in Line 4.

**Answer: A**

**Q: 36 Given:**

- 1. interface DoStuff2 {
- 2. float getRange(int low, int high); }
- 3.
- 4. interface DoMore {
- 5. float getAvg(int a, int b, int c); }

- 6.
7. abstract class DoAbstract implements DoStuff2, DoMore { }
- 8.
9. class DoStuff implements DoStuff2 {
10. public float getRange(int x, int y) { return 3.14f; } }
- 11.
12. interface DoAll extends DoMore {
13. float getAvg(int a, int b, int c, int d); }

**What is the result?**

- A. The file will compile without error.
- B. Compilation fails. Only line 7 contains an error.
- C. Compilation fails. Only line 12 contains an error.
- D. Compilation fails. Only line 13 contains an error.
- E. Compilation fails. Only lines 7 and 12 contain errors.
- F. Compilation fails. Only lines 7 and 13 contain errors.
- G. Compilation fails. Lines 7, 12, and 13 contain errors.

**Answer: A**

**Q: 37 Click the Exhibit button.**

**Given:**

25. A a = new A();

26. System.out.println(a.doit(4, 5));

**What is the result?**

```
1. public class A {  
2.     public String doit(int x, int y) {  
3.         return "a";  
4.     }  
5.  
6.     public String doit(int... vals) {  
7.         return "b";  
8.     }  
9. }
```

- A. Line 26 prints "a" to System.out.
- B. Line 26 prints "b" to System.out.
- C. An exception is thrown at line 26 at runtime.
- D. Compilation of class A will fail due to an error in line 6.

**Answer: A**

**Q: 38 Given:**

```
1. class TestA {  
2. public void start() { System.out.println("TestA"); }  
3. }  
  
4. public class TestB extends TestA {  
5. public void start() { System.out.println("TestB"); }  
6. public static void main(String[] args) {  
7. ((TestA)new TestB()).start();  
8. }  
9. }
```

**What is the result?**

- A. TestA
- B. TestB
- C. Compilation fails.
- D. An exception is thrown at runtime.

**Answer: B**

**Q: 39 Given:**

```
1. interface TestA { String toString(); }  
2. public class Test {
```



```
3. public static void main(String[] args) {  
4. System.out.println(new TestA() {  
5. public String toString() { return "test"; }  
6. });  
7. }  
8. }
```

**What is the result?**

- A. test
- B. null
- C. An exception is thrown at runtime.
- D. Compilation fails because of an error in line 1.
- E. Compilation fails because of an error in line 4.
- F. Compilation fails because of an error in line 5.

**Answer: A**

**Q: 40 Given:**

```
11. public class ItemTest {  
12. private final int id;  
13. public ItemTest(int id) { this.id = id; }  
14. public void updateId(int newId) { id = newId; }  
15.  
16. public static void main(String[] args) {  
17. ItemTest fa = new ItemTest(42);  
18. fa.updateId(69);  
19. System.out.println(fa.id);  
20. }
```

21. }

**What is the result?**

- A. Compilation fails.
- B. An exception is thrown at runtime.
- C. The attribute id in the Item object remains unchanged.
- D. The attribute id in the Item object is modified to the new value.
- E. A new Item object is created with the preferred value in the id attribute.

**Answer: A**

**Q: 41 Given:**

```
10. class One {  
11. void foo() { }  
12. }  
13. class Two extends One {  
14. //insert method here  
15. }
```

**Which three methods, inserted individually at line 14, will correctly complete class Two? (Choose three.)**

- A. `int foo() { /* more code here */ }`
- B. `void foo() { /* more code here */ }`
- C. `public void foo() { /* more code here */ }`
- D. `private void foo() { /* more code here */ }`
- E. `protected void foo() { /* more code here */ }`

**Answer: B, C, E**

**Q: 42 Given:**

```
10. interface Data { public void load(); }
```

**11. abstract class Info { public abstract void load(); }**

**Which class correctly uses the Data interface and Info class?**

A. public class Employee extends Info implements Data {

public void load() { /\*do something\*/ }

}

B. public class Employee implements Info extends Data {

public void load() { /\*do something\*/ } }

C. public class Employee extends Info implements Data

public void load(){ /\*do something\*/ }

public void Info.load(){ /\*do something\*/ }

}

D. public class Employee implements Info extends Data {

public void Data.load(){ /\*do something\*/ }

public void load(){ /\*do something\*/ }

}

E. public class Employee implements Info extends Data {

public void load(){ /\*do something\*/ }

public void Info.load(){ /\*do something\*/ }

}

F. public class Employee extends Info implements Data{

public void Data.load() { /\*do something\*/ }

public void Info.load() { /\*do something\*/ } }

**Answer: A**

**Q: 43 Given:**

**11. public abstract class Shape {**

**12. int x;**

```
13. int y;  
14. public abstract void draw();  
15. public void setAnchor(int x, int y) {  
16. this.x = x;  
17. this.y = y;  
18. }  
19. }
```

and a class Circle that extends and fully implements the Shape class.

**Which is correct?**

- A. Shape s = new Shape();  
s.setAnchor(10,10);  
s.draw();
- B. Circle c = new Shape();  
c.setAnchor(10,10);  
c.draw();
- C. Shape s = new Circle();  
s.setAnchor(10,10);  
s.draw();
- D. Shape s = new Circle();  
s->setAnchor(10,10);  
s->draw();
- E. Circle c = new Circle();  
c.Shape.setAnchor(10,10);  
c.Shape.draw();

**Answer: C**

**Q: 44 Click the Exhibit button.**

What is the result?

```
11. class Person {
12.     String name = "No name";
13.     public Person(String nm) { name = nm; }
14. }
15.
16. class Employee extends Person {
17.     String empID = "0000";
18.     public Employee(String id) : empID =
19.     id; }
20. }
21. public class EmployeeTest {
22.     public static void main(String[] args)
23.     {
24.         Employee e = new Employee("4321");
25.         System.out.println(e.empID);
26.     }
```

- A. 4321
- B. 0000
- C. An exception is thrown at runtime.
- D. Compilation fails because of an error in line 18.

**Answer: D**

**Q: 45 Given:**

- 10. interface Foo {}
- 11. class Alpha implements Foo {}
- 12. class Beta extends Alpha {}
- 13. class Delta extends Beta {
- 14. public static void main( String[] args ) {
- 15. Beta x = new Beta();
- 16. // insert code here
- 17. }
- 18. }

Which code, inserted at line 16, will cause a java.lang.ClassCastException?

- A. Alpha a = x;

- B. Foo f = (Delta)x;
- C. Foo f = (Alpha)x;
- D. Beta b = (Beta)(Alpha)x;

**Answer: B**

**Q: 46 Given:**

- 1. class SuperClass {
- 2. public A getA() {
- 3. return new A();
- 4. }
- 5. }
- 6. class SubClass extends SuperClass {
- 7. public B getA(){
- 8. return new B();
- 9. }
- 10. }

**Which statement is true?**

- A. Compilation will succeed if A extends B.
- B. Compilation will succeed if B extends A.
- C. Compilation will always fail because of an error in line 7.
- D. Compilation will always fail because of an error in line 8.

**Answer: B**

**Q: 47 Given:**

- 11. static class A {
- 12. void process() throws Exception { throw new Exception(); }

```
13. }  
14. static class B extends A {  
15. void process() { System.out.println("B"); }  
16. }  
17. public static void main(String[] args) {  
18. new B().process();  
19. }
```

**What is the result?**

- A. B
- B. The code runs with no output.
- C. Compilation fails because of an error in line 12.
- D. Compilation fails because of an error in line 15.
- E. Compilation fails because of an error in line 18.

**Answer: A**

**Q: 48 Given:**

```
11. class ClassA {}  
12. class ClassB extends ClassA {}  
13. class ClassC extends ClassA {}
```

**and:**

```
21. ClassA p0 = new ClassA();  
22. ClassB p1 = new ClassB();  
23. ClassC p2 = new ClassC();  
24. ClassA p3 = new ClassB();  
25. ClassA p4 = new ClassC();
```

**Which three are valid? (Choose three.)**

- A. p0 = p1;

- B. p1 = p2;
- C. p2 = p4;
- D. p2 = (ClassC)p1;
- E. p1 = (ClassB)p3;
- F. p2 = (ClassC)p4;

**Answer: A, E, F**

**Q: 49 Given:**

- 11. abstract class Vehicle { public int speed() { return 0; }
- 12. class Car extends Vehicle { public int speed() { return 60; }
- 13. class RaceCar extends Car { public int speed() { return 150; }
- ...
- 21. RaceCar racer = new RaceCar();
- 22. Car car = new RaceCar();
- 23. Vehicle vehicle = new RaceCar();
- 24. System.out.println(racer.speed() + ", " + car.speed()
- 25. + ", " + vehicle.speed());

**What is the result?**

- A. 0, 0, 0
- B. 150, 60, 0
- C. Compilation fails.
- D. 150, 150, 150
- E. An exception is thrown at runtime.

**Answer: D**



**Q: 50** Given code in separate source files:

```
10. public class Foo {  
11. public int a;  
12. public Foo() { a = 3; }  
13. public void addFive() { a += 5;}  
14. } and: 20. public class Bar extends Foo {  
21. public int a;  
22. public Bar() { a = 8; }  
23. public void addFive() { this.a += 5; }  
24. } invoked with:  
30. Foo foo = new Bar();  
31. foo.addFive();  
32. System.out.println("Value: " + foo.a);
```

**What is the result?**

- A. Value: 3
- B. Value: 8
- C. Value: 13
- D. Compilation fails.
- E. The code runs with no output.
- F. An exception is thrown at runtime.

**Answer: A**

**Q: 51** Which Man class properly represents the relationship "Man has a best friend who is a Dog"?

- A. class Man extends Dog { }
- B. class Man implements Dog { }

- C. class Man { private BestFriend dog; }
- D. class Man { private Dog bestFriend; }
- E. class Man { private Dog<bestFriend>; }
- F. class Man { private BestFriend<dog>; }

**Answer: D**

**Q: 52 Given:**

- 10. interface Foo { int bar(); }
- 11. public class Sprite {
- 12. public int fubar( Foo foo ) { return foo.bar(); }
- 13. public void testFoo() {
- 14. fubar(
- 15. // insert code here
- 16. );
- 17. }
- 18. }

**Which code, inserted at line 15, allows the class Sprite to compile?**

- A. Foo { public int bar() { return 1; } }
- B. new Foo { public int bar() { return 1; } }
- C. new Foo() { public int bar() { return 1; } }
- D. new class Foo { public int bar() { return 1; } }

**Answer: C**

**Q: 53 Click the Task button.**

Given: 

```
public class Doubler {
    public static int doubleMe( Holder h) {
        return h.getAmount() * 2;
    }
}
```

and: 

```
public class Holder {
    int amount = 10;
    public void doubleAmount(){ amount = Doubler.doubleMe( this );}
    public int getAmount(){ return amount;}
    //more code here
}
```

Place the code fragments in position to reduce the coupling between Doubler and Holder.

```
public class Doubler {
    public static int doubleMe( Place here h) {
        return Place here * 2;
    }
}

public class Holder {
    int amount = 10;
    public void doubleAmount(){ amount = Doubler.doubleMe( Place here );}
    public int getAmount(){ return amount;}
    //more code here
}
```

#### Code Fragments

void	Holder	int	Doubler
h.getAmount()	h	this	amount

Done

**Solution:**

1. int   2. h   3. amount.

**Q: 54** Given classes defined in two different files:

1. package packageA;

2. public class Message {

3. String getText() { return "text"; }

4. }

and:

1. package packageB;

```
2. public class XMLMessage extends packageA.Message {  
3. String getText() { return "<msg>text</msg>";}  
4. public static void main(String[] args) {  
5. System.out.println(new XMLMessage().getText());  
6. }  
7. }
```

What is the result of executing XMLMessage.main?

- A. text
- B. An exception is thrown at runtime.
- C. Compilation fails because of an error in line 2 of XMLMessage.
- D. Compilation fails because of an error in line 3 of XMLMessage.

Answer: D

Q: 55 Given:

```
1. public class A {  
2. public void doit() {  
3. }  
4. public String doit() {  
5. return "a";  
6. }  
7. public double doit(int x) {  
8. return 1.0;  
9. }  
10. }
```

What is the result?

- A. An exception is thrown at runtime.
- B. Compilation fails because of an error in line 7.
- C. Compilation fails because of an error in line 4.
- D. Compilation succeeds and no runtime errors with class A occur.

**Answer: C**

**Question: 56**

**Click the Exhibit button.**

1. `public class GoTest {`
2. `public static void main(String[] args) {`
3. `Sente a = new Sente(); a.go();`
4. `Goban b = new Goban(); b.go();`
5. `Stone c = new Stone(); c.go();`
6. `}`
7. `}`
- 8.
9. `class Sente implements Go {`
10. `public void go() { System.out.println("go in Sente."); }`
11. `}`
- 12.
13. `class Goban extends Sente {`
14. `public void go() { System.out.println("go in Goban"); }`
15. `}`
- 16.
17. `class Stone extends Goban implements Go { }`
- 18.
19. `interface Go { public void go(); }`

**What is the result?**

A. go in Goban

go in Sente

go in Sente

B. go in Sente

go in Sente

go in Goban

C. go in Sente

go in Goban

go in Goban

D. go in Goban

go in Goban

go in Sente

E. Compilation fails because of an error in line 17.

**Answer: C**

**Question: 57**

**Click the Exhibit button.**

```
1. public class Employee {  
2. String name;  
3. double baseSalary;  
4. Employee(String name, double baseSalary) {  
5. this.name = name;  
6. this.baseSalary = baseSalary;  
7. }
```

8. }

And:

```
1. public class Salesperson extends Employee {  
2. double commission;  
3. public Salesperson(String name, double baseSalary,  
4. double commission) {  
5. // insert code here  
6. }  
7. }
```

Which code, inserted at line 7, completes the Salesperson constructor?

A. this.commission = commission;

B. superb();

commission = commission;

C. this.commission = commission;

superb();

D. super(name, baseSalary);

this.commission = commission;

E. super();

this.commission = commission;

F. this.commission = commission;

super(name, baseSalary);

**Answer: D**

**Question: 58**

**Given:**

```
1. public class Team extends java.util.LinkedList {  
2. public void addPlayer(Player p) {  
3. add(p);  
4. }  
5. public void compete(Team opponent) { /* more code here */ }  
6. }  
7. class Player { /* more code here */ }
```

**Which two are true? (Choose two.)**

- A. This code will compile.
- B. This code demonstrates proper design of an is-a relationship.
- C. This code demonstrates proper design of a has-a relationship.
- D. A Java programmer using the Team class could remove Player objects from a Team object.

**Answer: AD**

**Question: 59**

**Given:**

```
5. class Atom {  
6. Atom() { System.out.print("atom "); }  
7. }  
8. class Rock extends Atom {  
9. Rock(String type) { System.out.print(type); }  
10. }  
11. public class Mountain extends Rock {  
12. Mountain() {  
13. super("granite ");
```



```
14. new Rock("granite ");  
15. }  
16. public static void main(String[] a) { new Mountain(); }  
17. }
```

**What is the result?**

- A. Compilation fails.
- B. atom granite
- C. granite granite
- D. atom granite granite
- E. An exception is thrown at runtime.
- F. atom granite atom granite

**Answer:** F

**Question: 60**

**Given:**

```
5. class Building { }  
6. public class Barn extends Building {  
7. public static void main(String[] args) {  
8. Building build1 = new Building();  
9. Barn barn1 = new Barn();  
10. Barn barn2 = (Barn) build1;  
11. Object obj1 = (Object) build1;  
12. String str1 = (String) build1;  
13. Building build2 = (Building) barn1;  
14. }  
15. }
```

**Which is true?**

- A. If line 10 is removed, the compilation succeeds.
- B. If line 11 is removed, the compilation succeeds.
- C. If line 12 is removed, the compilation succeeds.
- D. If line 13 is removed, the compilation succeeds.
- E. More than one line must be removed for compilation to succeed.

**Answer:** C

**Question: 61**

A company has a business application that provides its users with many different reports receivables reports, payables reports, revenue projects, and so on. The company has just purchased some new, state-of-the-art, wireless printers, and a programmer has been assigned the task of enhancing all of the reports to use not only the company's old printers, but the new wireless printers as well. When the programmer starts looking into the application, the programmer discovers that because of the design of the application, it is necessary to make changes to each report to support the new printers. Which two design concepts most likely explain this situation? (Choose two.)

- A. Inheritance
- B. Low cohesion
- C. Tight coupling
- D. High cohesion
- E. Loose coupling
- F. Object immutability

**Answer:** B,C

**Question: 62**

**Given:**

```
31. class Foo {  
32.     public int a = 3;  
33.     public void addFive() { a += 5; System.out.print("f "); }  
34. }
```

```
35. class Bar extends Foo {  
36. public int a = 8;  
37. public void addFive() { this.a += 5; System.out.print("b "); }  
38. }
```

Invoked with:

```
Foo f = new Bar();
```

```
f.addFive();
```

```
System.out.println(f.a);
```

What is the result?

- A. b 3
- B. b 8
- C. b 13
- D. f 3
- E. f 8
- F. f 13
- G. Compilation fails.
- H. An exception is thrown at runtime.

**Answer:** A

**Question: 63**

A company that makes Computer Assisted Design (CAD) software has, within its application, some utility classes that are used to perform 3D rendering tasks. The company's chief scientist has just improved the performance of one of the utility classes' key rendering algorithms, and has assigned a programmer to replace the old algorithm with the new algorithm. When the programmer begins researching the utility classes, she is happy to discover that the algorithm to be replaced exists in only one class. The programmer reviews that class's API, and replaces the old algorithm with the new

algorithm, being careful that her changes adhere strictly to the class's API. Once testing has begun, the programmer discovers that other classes that use the class she changed are no longer working properly. What design flaw is most likely the cause of these new bugs?

- A. Inheritance
- B. Tight coupling
- C. Low cohesion
- D. High cohesion
- E. Loose coupling
- F. Object immutability

**Answer:** B

**Question: 64**

**Given:**

- 5. class Thingy { Meter m = new Meter(); }
- 6. class Component { void go() { System.out.print("c"); } }
- 7. class Meter extends Component { void go() { System.out.print("m"); } }
- 8.
- 9. class DeluxeThingy extends Thingy {
- 10. public static void main(String[] args) {
- 11. DeluxeThingy dt = new DeluxeThingy();
- 12. dt.m.go();
- 13. Thingy t = new DeluxeThingy();
- 14. t.m.go();
- 15. }
- 16. }

**Which two are true? (Choose two.)**

- A. The output is mm.
- B. The output is mc.
- C. Component is-a Meter.
- D. Component has-a Meter.
- E. DeluxeThingy is-a Component.
- F. DeluxeThingy has-a Component.

**Answer:** A,F

**Question: 65**

**Given:**

1. class X {
2. X() { System.out.print(1); }
3. X(int x) {
4. this(); System.out.print(2);
5. }
6. }
7. public class Y extends X {
8. Y() { super(6); System.out.print(3); }
9. Y(int y) {
10. this(); System.out.println(4);
11. }
12. public static void main(String[] a) { new Y(5); }
13. }

**What is the result?**

- A. 13
- B. 134
- C. 1234

D. 2134

E. 2143

F. 4321

**Answer:** C

**Question: 66**

**Given:**

2. `public class Hi {`

3. `void m1() {}`

4. `protected void m2 {}`

5. `}`

6. `class Lois extends Hi {`

7. `// insert code here`

8. `}`

**Which four code fragments, inserted independently at line 7, will compile? (Choose four.)**

A. `public void m1() {}`

B. `protected void m1() {}`

C. `private void m1() {}`

D. `void m2() {}`

E. `public void m2() {}`

F. `protected void m2() {}`

G. `private void m2() {}`

**Answer:** A,B,E,F

**Question: 67**

**Given:**

11. `public interface A { public void m1(); }`
- 12.
13. `class B implements A { }`
14. `class C implements A { public void m1() { } }`
15. `class D implements A { public void m1(int x) { } }`
16. `abstract class E implements A { }`
17. `abstract class F implements A { public void m1() { } }`
18. `abstract class G implements A { public void m1(int x) { } }`

**What is the result?**

- A. Compilation succeeds.
- B. Exactly one class does NOT compile.
- C. Exactly two classes do NOT compile.
- D. Exactly four classes do NOT compile.
- E. Exactly three classes do NOT compile.

**Answer:** C

**Question: 68**

**Given:**

3. `interface Fish { }`
4. `class Perch implements Fish { }`
5. `class Walleye extends Perch { }`
6. `class Bluegill { }`
7. `public class Fisherman {`
8. `public static void main(String[] args) {`
9. `Fish f = new Walleye();`
10. `Walleye w = new Walleye();`
11. `Bluegill b = new Bluegill();`

```
12. if(f instanceof Perch) System.out.print("f-p ");
13. if(w instanceof Fish) System.out.print("w-f ");
14. if(b instanceof Fish) System.out.print("b-f ");
15. }
16. }
```

**What is the result?**

- A. w-f
- B. f-p w-f
- C. w-f b-f
- D. f-p w-f b-f
- E. Compilation fails.
- F. An exception is thrown at runtime.

**Answer:** B

**Question: 69**

```
11. public interface A111 {
12. String s = "yo";
13. public void method1();
14. }
17. interface B {}
20. interface C extends A111, B {
21. public void method1();
22. public void method1(int x);
23. }
```

**What is the result?**

- A. Compilation succeeds.
- B. Compilation fails due to multiple errors.
- C. Compilation fails due to an error only on line 20.



- D. Compilation fails due to an error only on line 21.
- E. Compilation fails due to an error only on line 22.
- F. Compilation fails due to an error only on line 12.

**Answer: A**

**Question: 70**

```
10. import java.io.*;
11. class Animal {
12. Animal() { System.out.print("a"); }
13. }
14. class Dog extends Animal implements Serializable {
15. Dog() { System.out.print("d"); }
16. }
17. public class Beagle extends Dog { }
```

If an instance of class Beagle is created, then Serialized, then deSerialized, what is the result?

- A. ad
- B. ada
- C. add
- D. adad
- E. Compilation fails.
- F. An exception is thrown at runtime.

**Answer: B**

**Question: 71**

A team of programmers is involved in reviewing a proposed design for a new utility class. After some discussion, they realize that the current design allows other classes to

access methods in the utility class that should be accessible only to methods within the utility class itself. What design issue has the team discovered?

- A. Tight coupling
- B. Low cohesion
- C. High cohesion
- D. Loose coupling
- E. Weak encapsulation
- F. Strong encapsulation

**Answer:** E

**Question: 72**

**Given that: Gadget has-a Sprocket and Gadget has-a Spring and Gadget is-a Widget and Widget has-a Sprocket Which two code fragments represent these relationships? (Choose two.)**

- A. `class Widget { Sprocket s; } class Gadget extends Widget { Spring s; }`
- B. `class Widget { } class Gadget extends Widget { Spring s1; Sprocket s2; }`
- C. `class Widget { Sprocket s1; Spring s2; } class Gadget extends Widget { }`
- D. `class Gadget { Spring s; } class Widget extends Gadget{ Sprocket s; }`
- E. `class Gadget { } class Widget extends Gadget{ Sprocket s1; Spring s2; }`
- F. `class Gadget { Spring s1; Sprocket s2; } class Widget extends Gadget{ }`

**Answer:** A,C

**Question: 73**

- 11. `class Alpha {`
- 12. `public void foo() { System.out.print("Afoo "); }`
- 13. `}`
- 14. `public class Beta extends Alpha {`

```
15. public void foo() { System.out.print("Bfoo "); }  
16. public static void main(String[] args) {  
17. Alpha a = new Beta();  
18. Beta b = (Beta)a;  
19. a.foo(); 20. b.foo();  
21. }  
22. }
```

**What is the result?**

- A. Afoo Afoo
- B. Afoo Bfoo
- C. Bfoo Afoo
- D. Bfoo Bfoo
- E. Compilation fails.
- F. An exception is thrown at runtime.

**Answer:** D

**Question:** 74

Click the Exhibit button.

```
11. class Payload {  
12. private int weight;  
13. public Payload(int wt) { weight = wt; }  
13. public void setWeight(mt w) { weight = w; }  
15. public String toString { return Integer.toString(weight); }  
16. }  
17.  
18. public class TestPayload {  
19. static void changePayload(Payload p) {
```

```
20. /* insert code here */
21. }
22.
23. public static void main(String[] args) {
24.     Payload p = new Payload();
25.     p.setWeight(1024);
26.     changePayload(p);
27.     System.out.println("The value of p is " + p);
28. }
29. }
```

Which statement, placed at line 20, causes the code to print "The value of p is 420."?

- A. p.setWeight(420);
  - B. p.changePayload(420);
  - C. p = new Payload(420);
  - D. Payload.setWeight(420);
  - E. p = Payload.setWeight(420);
  - F. p = new Payload();
- p.setWeight(420);

**Answer: A**

## 9. Multi-Threading

Q: 01 Click the Exhibit button.

What is the result?

```
1.  class Computation extends Thread {
2.
3.      private int num;
4.      private boolean isComplete;
5.      private int result;
6.
7.      public Computation(int num) { this.num
8.  = num; }
9.
10.     public synchronized void run() {
11.         result = num * 2;
12.         isComplete = true;
13.         notify();
14.     }
15.
16.     public synchronized int getResult() {
17.         while (!isComplete) {
18.             try {
19.                 wait();
20.             } catch (InterruptedException e)
21.             {}
22.         }
23.         return result;
24.     }
25.
26.     public static void main(String[] args)
27.     {
28.         Computation[] computations = new
29. Computation[4];
30.         for (int i = 0; i <
31. computations.length; i++) {
32.             computations[i] = new
33. Computation(i);
34.             computations[i].start();
35.         }
36.         for (Computation c : computations)
37.             System.out.print(c.getResult() + "
38. ");
39.     }
40. }
```

- A. The code will deadlock.
- B. The code may run with no output.
- C. An exception is thrown at runtime.
- D. The code may run with output "0 6".

E. The code may run with output "2 0 6 4".

F. The code may run with output "0 2 4 6".

**Answer: F**

**Q: 02 Given:**

```
1. public class Threads2 implements Runnable {  
2.  
3. public void run() {  
4. System.out.println("run.");  
5. throw new RuntimeException("Problem");  
6. }  
7. public static void main(String[] args) {  
8. Thread t = new Thread(new Threads2());  
9. t.start();  
10. System.out.println("End of method.");  
11. }  
12. }
```

**Which two can be results? (Choose two.)**

A. java.lang.RuntimeException: Problem

B. run.

java.lang.RuntimeException: Problem

C. End of method.

java.lang.RuntimeException: Problem

D. End of method.

run.

java.lang.RuntimeException: Problem

E. run.

java.lang.RuntimeException: Problem

End of method.

**Answer: D, E**

**Q: 03 Given:**

```
1. public class TestSeven extends Thread {  
2.     private static int x;  
3.     public synchronized void doThings() {  
4.         int current = x;  
5.         current++;  
6.         x = current;  
7.     }  
8.     public void run() {  
9.         doThings();  
10.    }  
11. }
```

**Which statement is true?**

- A. Compilation fails.
- B. An exception is thrown at runtime.
- C. Synchronizing the run() method would make the class thread-safe.
- D. The data in variable "x" are protected from concurrent access problems.
- E. Declaring the doThings() method as static would make the class thread-safe.
- F. Wrapping the statements within doThings() in a synchronized(new Object()) { } block would make the class thread-safe.

**Answer: E**

**Q: 04 Given:**

```
1. public class Threads3 implements Runnable {  
2.     public void run() {  
3.         System.out.print("running");  
4.     }  
5.     public static void main(String[] args) {  
6.         Thread t = new Thread(new Threads3());  
7.         t.run();  
8.         t.run();  
9.         t.start();  
10.    }  
11. }
```

**What is the result?**

- A. Compilation fails.
- B. An exception is thrown at runtime.
- C. The code executes and prints "running".
- D. The code executes and prints "runningrunning".
- E. The code executes and prints "runningrunningrunning".

**Answer: E**

**Q: 05 Given:**

```
public class NamedCounter {  
    private final String name;  
    private int count;  
    public NamedCounter(String name) { this.name = name; }
```



```
public String getName() { return name; }  
  
public void increment() { count++; }  
  
public int getCount() { return count; }  
  
public void reset() { count = 0; }  
  
}
```

Which three changes should be made to adapt this class to be used safely by multiple threads?  
(Choose

three.)

- A. declare reset() using the synchronized keyword
- B. declare getName() using the synchronized keyword
- C. declare getCount() using the synchronized keyword
- D. declare the constructor using the synchronized keyword
- E. declare increment() using the synchronized keyword

**Answer: A, C, E**

**Q: 06 Given:**

```
7. void waitForSignal() {  
  
8. Object obj = new Object();  
  
9. synchronized (Thread.currentThread()) {  
  
10. obj.wait();  
  
11. obj.notify();  
  
12. }  
  
13. }
```

Which statement is true?

- A. This code may throw an InterruptedException.
- B. This code may throw an IllegalStateException.
- C. This code may throw a TimeoutException after ten minutes.
- D. This code will not compile unless "obj.wait()" is replaced with "((Thread) obj).wait()".

E. Reversing the order of `obj.wait()` and `obj.notify()` may cause this method to complete normally.

F. A call to `notify()` or `notifyAll()` from another thread may cause this method to complete normally.

**Answer: B**

**Q: 07 Which two code fragments will execute the method `doStuff()` in a separate thread? (Choose two.)**

A. `new Thread() {  
 public void run() { doStuff(); }  
};`

B. `new Thread() {  
 public void start() { doStuff(); }  
};`

C. `new Thread() {  
 public void start() { doStuff(); }  
}.run();`

D. `new Thread() {  
 public void run() { doStuff(); }  
}.start();`

E. `new Thread(new Runnable() {  
 public void run() { doStuff(); }  
}).run();`

F. `new Thread(new Runnable() {  
 public void run() { doStuff(); }  
}).start();`

**Answer: D, F**

**Q: 08 Given:**

```
1. public class TestOne implements Runnable {  
2. public static void main (String[] args) throws Exception {  
3. Thread t = new Thread(new TestOne());  
4. t.start();  
5. System.out.print("Started");  
6. t.join();  
7. System.out.print("Complete");  
8. }  
9. public void run() {  
10. for (int i = 0; i < 4; i++) {  
11. System.out.print(i);  
12. }  
13. }  
14. }
```

**What can be a result?**

- A. Compilation fails.
- B. An exception is thrown at runtime.
- C. The code executes and prints "StartedComplete".
- D. The code executes and prints "StartedComplete0123".
- E. The code executes and prints "Started0123Complete".

**Answer: E**

**Q: 09Given:**

```
1. public class TestOne {  
2. public static void main (String[] args) throws Exception {  
3. Thread.sleep(3000);
```

4. `System.out.println("sleep");`

5. `}`

6. `}`

**What is the result?**

- A. Compilation fails.
- B. An exception is thrown at runtime.
- C. The code executes normally and prints "sleep".
- D. The code executes normally, but nothing is printed.

**Answer: C**

**Q: 10 Given:**

11. `public class Test {`

12. `public enum Dogs {collie, harrier, shepherd};`

13. `public static void main(String [] args) {`

14. `Dogs myDog = Dogs.shepherd;`

15. `switch (myDog) {`

16. `case collie:`

17. `System.out.print("collie ");`

18. `case default:`

19. `System.out.print("retriever ");`

20. `case harrier:`

21. `System.out.print("harrier ");`

22. `}`

23. `}`

24. `}`

**What is the result?**

- A. harrier

- B. shepherd
- C. retriever
- D. Compilation fails.
- E. retriever harrier
- F. An exception is thrown at runtime.

**Answer: D**

**Q: 11 Given:**

```
11. Runnable r = new Runnable() {  
12. public void run() {  
13. System.out.print("Cat");  
14. }  
15. };  
16. Thread t = new Thread(r) {  
17. public void run() {  
18. System.out.print("Dog");  
19. }  
20. };  
21. t.start();
```

**What is the result?**

- A. Cat
- B. Dog
- C. Compilation fails.
- D. The code runs with no output.
- E. An exception is thrown at runtime.

**Answer: B**

Q: 12 Given:

```
1. public class Threads4 {  
2. public static void main (String[] args) {  
3. new Threads4().go();  
4. }  
5. public void go() {  
6. Runnable r = new Runnable() {  
7. public void run() {  
8. System.out.print("foo");  
9. }  
10. };  
11. Thread t = new Thread(r);  
12. t.start();  
13. t.start();  
14. }  
15. }
```

What is the result?

- A. Compilation fails.
- B. An exception is thrown at runtime.
- C. The code executes normally and prints "foo".
- D. The code executes normally, but nothing is printed.

Answer: B

Q: 13 Given:

```
1. public class TestFive {  
2. private int x;
```

```
3. public void foo() {  
4.     int current = x;  
5.     x = current + 1;  
6. }  
7. public void go() {  
8.     for(int i = 0; i < 5; i++) {  
9.         new Thread() {  
10.            public void run() {  
11.                foo();  
12.                System.out.print(x + ", ");  
13.            }.start();  
14.        } }  
15. }
```

**Which two changes, taken together, would guarantee the output: 1, 2, 3, 4, 5, ? (Choose two.)**

- A. move the line 12 print statement into the foo() method
  - B. change line 7 to public synchronized void go() {
  - C. change the variable declaration on line 2 to private volatile int x;
  - D. wrap the code inside the foo() method with a synchronized( this ) block
  - E. wrap the for loop code inside the go() method with a synchronized block synchronized(this)
- ```
{ // for loop code here }
```

**Answer: A, D**

Q: 14 Click the Task button.

```

Given: 10. Runnable r = new Runnable() {
        11.     public void run() {
        12.         try {
        13.             Thread.sleep(1000);
        14.         } catch (InterruptedException e) {
        15.             System.out.println("interrupted");
        16.         }
        17.         System.out.println("ran");
        18.     }
        19. };
        20. Thread t = new Thread(r);
        21. t.start();
        22. System.out.println("started");
        23. t.sleep(2000);
        24. System.out.println("interrupting");
        25. t.interrupt();
        26. System.out.println("ended");

```

Assume that sleep(n) executes in exactly n milliseconds, and all other code executes in an insignificant amount of time.

Place the fragments in the output area to show the result of running this code.

| Output     | Fragments            |
|------------|----------------------|
| Place here | interrupted          |
| Place here | ran                  |
| Place here | started              |
| Place here | interrupting         |
| Place here | ended                |
| Place here | InterruptedException |
|            | (no more output)     |

Done

**Solution:**

- 1.started
- 2.ran
- 3.interrupting
- 4.ended
- 5.(no more out put)

15 Click the Exhibit button.

What is the output if the main() method is run?



Given:

```
10. public class Starter extends Thread {
11.     private int x = 2;
12.     public static void main(String[] args)
13.     throws Exception {
14.         new Starter().makeItSo();
15.     }
16.     public Starter() {
17.         x = 5;
18.         start();
19.     }
20.     public void makeItSo() throws
21.     Exception {
22.         join();
23.         x = x - 1;
24.         System.out.println(x);
25.     }
26.     public void run() { x *= 2; }
```

- A. 4
- B. 5
- C. 8
- D. 9
- E. Compilation fails.
- F. An exception is thrown at runtime.
- G. It is impossible to determine for certain.

**Answer: D**

**Q: 16 Click the Task button.**

Place a Class on each method that is declared in the class.

| Method Name | Class            |
|-------------|------------------|
| run()       | java.lang.Object |
| wait()      | java.lang.Thread |
| notify()    |                  |
| sleep()     |                  |
| start()     |                  |
| join()      |                  |

Done

**Solution:**

java.lang.Object ----- wait(); notify();

java.lang.Thread-----run(), sleep(); start(); join ()

**Q: 17 Given:**

foo and bar are public references available to many other threads. foo refers to a Thread and bar is an

Object. The thread foo is currently executing bar.wait().

From another thread, what provides the most reliable way to ensure that foo will stop executing wait()?

- |                     |                     |
|---------------------|---------------------|
| A. foo.notify();    | B. bar.notify();    |
| C. foo.notifyAll(); | D. Thread.notify(); |
| E. bar.notifyAll(); | F. Object.notify(); |

**Answer: E**

**Q: 18 Given:**

```
1. public class MyLogger {  
2.     private StringBuilder logger = new StringBuuilder();  
3.     public void log(String message, String user) {  
4.         logger.append(message);  
5.         logger.append(user);  
6.     }  
7. }
```

The programmer must guarantee that a single MyLogger object works properly for a multi-threaded system.

**How must this code be changed to be thread-safe?**

- A. synchronize the log method
- B. replace StringBuilder with StringBuffer
- C. replace StringBuilder with just a String object and use the string concatenation (+) within the log method
- D. No change is necessary, the current MyLogger code is already thread-safe.

**Answer: A**

**Q: 19 Given:**

```
1. public class TestSeven extends Thread {  
2.     private static int x;  
3.     public synchronized void doThings() {  
4.         int current = x;  
5.         current++;
```

6. x = current;

7. }

8. public void run() {

9. doThings();

10. }

11. }

**Which statement is true?**

A. Compilation fails.

B. An exception is thrown at runtime.

C. Synchronizing the run() method would make the class thread-safe.

D. The data in variable "x" are protected from concurrent access problems.

E. Declaring the doThings() method as static would make the class thread-safe.

F. Wrapping the statements within doThings() in a synchronized(new Object()) { } block would make the class thread-safe.

**Answer: E**

**Q:20 Click the Exhibit button.**

**Which two statements are true if this class is compiled and run? (Choose two.)**

```
1. import java.util.*;
2.
3. public class NameList {
4.     private List names = new ArrayList();
5.     public synchronized void add(String
name) { names.add(name); }
6.     public synchronized void printAll() {
7.         for (int i = 0; i < names.size();
i++) {
8.             System.out.print(names.get(i) + "
");
9.         }
10.    }
11.    public static void main(String[] args)
{
12.        final NameList sl = new NameList();
13.        for (int i = 0; i < 2; i++) {
14.            new Thread() {
15.                public void run() {
16.                    sl.add("A");
17.                    sl.add("B");
18.                    sl.add("C");
19.                    sl.printAll();
20.                }
21.            }.start();
22.        }
23.    }
24. }
```

A. An exception may be thrown at runtime.

- B. The code may run with no output, without exiting.
- C. The code may run with no output, exiting normally.
- D. The code may run with output "A B A B C C ", then exit.
- E. The code may run with output "A B C A B C A B C ", then exit.
- F. The code may run with output "A A A B C A B C C ", then exit.
- G. The code may run with output "A B C A A B C A B C ", then exit.

**Answer: E, G**

**Q: 21 Given:**

```
1. public class Threads5 {  
2.     public static void main (String[] args) {  
3.         new Thread(new Runnable() {  
4.             public void run() {  
5.                 System.out.print("bar");  
6.             }).start();  
7.         }  
8.     }
```

**What is the result?**

- A. Compilation fails.
- B. An exception is thrown at runtime.
- C. The code executes normally and prints "bar".
- D. The code executes normally, but nothing prints.

**Answer: C**

**Q: 22 Which three will compile and run without exception? (Choose three.)**

- A. private synchronized Object o;
- B. void go() {
- synchronized() { /\* code here \*/ }
- C. public synchronized void go() { /\* code here \*/ }
- D. private synchronized(this) void go() { /\* code here \*/ }
- E. void go() {
- synchronized(Object.class) { /\* code here \*/ }
- F. void go() {
- Object o = new Object();
- synchronized(o) { /\* code here \*/ }

**Answer: C, E, F**

**Q: 23 Click the Task button.**

Place the code elements in position so that the Flags2 class will compile and make appropriate use of the wait/notify mechanism.  
Note: You may reuse code elements

```
public class Flags2 {  
    private boolean isReady = false;  
  
    public Place here void produce() {  
        isReady = true;  
        Place here;  
    }  
  
    public Place here void consume() {  
        while (! isReady) {  
            try {  
                Place here;  
            } catch (Exception ex) { }  
        }  
        isReady = Place here;  
    }  
}
```

**Code Elements**

|              |                |             |             |
|--------------|----------------|-------------|-------------|
| synchronized | true           | false       | wait()      |
| volatile     | synchronized() | notifyAll() | synchronize |

**Solution:**

- 1.synchronized
- 2.notifyAll()
- 3.synchronized
- 4.wait()
- 5.false

**Q: 24 Click the Task button.**

Place the code elements into the class so that the code compiles and prints "Run. Run. doIt." in exactly that order. Note that there may be more than one correct solution.

```
public class TestTwo extends Thread {  
    public static void main (String[] a) throws Exception {  
        TestTwo t = new TestTwo();  
        t.start();  
          
          
    }   
    public void run() {  
        System.out.print("Run. ");  
    }  
    public void doIt() {  
        System.out.print("doIt. ");  
    }  
}
```

**Code Elements**

|                         |                        |                           |                     |                                     |
|-------------------------|------------------------|---------------------------|---------------------|-------------------------------------|
| <code>t.start();</code> | <code>t.join();</code> | <code>t.pause(10);</code> | <code>run();</code> | <input type="button" value="Done"/> |
| <code>t.run();</code>   | <code>t.doIt();</code> | <code>doIt();</code>      |                     |                                     |

**Solution:**

`t.join();`  
`t.run();`  
`t.doIt();`

Q: 25 Click the Exhibit button.

Which two are possible results? (Choose two.)

```
1. public class Threadsl {
2.     int x = 0;
3.     public class Runner implements Runnable
4.     {
5.         public void run() {
6.             int current = 0;
7.             for(int i = 0; i < 4; i++) {
8.                 current = x;
9.                 System.out.print(current + ", ");
10.                x = current + 2;
11.            }
12.        }
13.    }
14.    public static void main(String[] args) {
15.        new Threadsl().go();
16.    }
17.    public void go() {
18.        Runnable r1 = new Runner();
19.        new Thread(r1).start();
20.        new Thread(r1).start();
21.    }
22. }
23. }
```

- A. 0, 2, 4, 4, 6, 8, 10, 6,
- B. 0, 2, 4, 6, 8, 10, 2, 4,
- C. 0, 2, 4, 6, 8, 10, 12, 14,
- D. 0, 0, 2, 2, 4, 4, 6, 6, 8, 8, 10, 10, 12, 12, 14, 14,
- E. 0, 2, 4, 6, 8, 10, 12, 14, 0, 2, 4, 6, 8, 10, 12, 14,

Answer: A, C

Question: 26

Click the Exhibit button.



Given:

```
1. public class TwoThreads {  
2  
3. private static Object resource = new Object();  
4.  
5. private static void delay(long n) {  
6. try { Thread.sleep(n); }  
7. catch (Exception e) { System.out.print("Error "); }  
8. }  
9  
10. public static void main(String[] args) {  
11. System.out.print("StartMain ");  
12. new Thread1().start();  
13. delay(1000);  
14. Thread t2 = new Thread2();  
15. t2.start();  
16. delay(1000);  
17. t2.interrupt  
18. delay(1000);  
19. System.out.print("EndMain ");  
20. }  
21.  
22. static class Thread 1 extends Thread {  
23. public void run() {  
24. synchronized (resource) {  
25. System.out.print("Start1 ");  
26. delay(6000);
```

```
27. System.out.print("End1 ");
28. }
29. }
30. }
31.
32. static class Thread2 extends Thread {
33. public void run() {
34. synchronized (resource) {
35. System.out.print("Start2 ");
36. delay(2000);
37. System.out.print("End2 ");
38. }
39. }
40. }
41. }
```

Assume that sleep(n) executes in exactly n milliseconds, and all other code executes in an insignificant amount of time. What is the output if the main() method is run?

- A. Compilation fails.
- B. Deadlock occurs.
- C. StartMain Start1 Error EndMain End1
- D. StartMain Start1 EndMain End1 Start2 End2
- E. StartMain Start1 Error Start2 EndMain End2 End1
- F. StartMain Start1 Start2 Error End2 EndMain End1
- G. StartMain Start1 EndMain End1 Start2 Error End2

**Answer: G**

Question: 27

Click the Exhibit button.

```
10. public class Transfers {
11.     public static void main(String[] args) throws Exception {
12.         Record r1 = new Record();
13.         Record r2 = new Record();
14.         doTransfer(r1, r2, 5);
15.         doTransfer(r2, r1, 2);
16.         doTransfer(r1, r2, 1);
17.         // print the result
18.         System.out.println("r1 = " + r1.get() + ", r2=" + r2.get());
19.     }
20.     private static void doTransfer(
21.         final Record a, final Record b, final int amount) {
22.         Thread t = new Thread() {
23.             public void run() {
24.                 new Clerk().transfer(a, b, amount);
25.             }
26.         };
27.         t.start();
28.     }
29. }
30. class Clerk {
31.     public synchronized void transfer(Record a, Record b, int amount){
32.         synchronized (a) {
```

```
33. synchronized (b) {  
34. a.add(-amount);  
35. b.add(amount);  
36. }  
37. }  
38. }  
39. }  
40. class Record {  
41.int num=10;  
42. public int get() { return num; }  
43. public void add(int n) { num = num + n; }  
44. }
```

If Transfers.main() is run, which three are true? (Choose three.)

- A. The output may be "r1 = 6, r2 = 14".
- B. The output may be "r1 = 5, r2 = 15".
- C. The output may be "r1 = 8, r2 = 12".
- D. The code may run (and complete) with no output.
- E. The code may deadlock (without completing) with no output.
- F. M IllegalStateException or InterruptedException may be thrown at runtime.

**Answer: ABE**

**Question: 28**

**Which two statements are true? (Choose two.)**

- A. It is possible for more than two threads to deadlock at once.
- B. The JVM implementation guarantees that multiple threads cannot enter into a deadlocked state.

- C. Deadlocked threads release once their sleep() method's sleep duration has expired.
- D. Deadlocking can occur only when the wait(), notify(), and notifyAll() methods are used incorrectly.
- E. It is possible for a single-threaded application to deadlock if synchronized blocks are used incorrectly.
- F. If a piece of code is capable of deadlocking, you cannot eliminate the possibility of deadlocking by inserting invocations of Thread.yield().

**Answer:** A, F

**Question: 29**

**Given:**

```
11. class PingPong2 {  
12.     synchronized void hit(long n) {  
13.         for(int i = 1; i < 3; i++)  
14.             System.out.print(n + "-" + i + " ");  
15.     }  
16. }  
  
17. public class Tester implements Runnable {  
18.     static PingPong2 pp2 = new PingPong2();  
19.     public static void main(String[] args) {  
20.         new Thread(new Tester()).start();  
21.         new Thread(new Tester()).start();  
22.     }  
23.     public void run() { pp2.hit(Thread.currentThread().getId()); }  
24. }
```

**Which statement is true?**

- A. The output could be 5-1 6-1 6-2 5-2
- B. The output could be 6-1 6-2 5-1 5-2
- C. The output could be 6-1 5-2 6-2 5-1
- D. The output could be 6-1 6-2 5-1 7-1

**Answer: B**

**Question: 30**

**Given that t1 is a reference to a live thread, which is true?**

- A. The Thread.sleep() method can take t1 as an argument.
- B. The Object.notify() method can take t1 as an argument.
- C. The Thread.yield() method can take t1 as an argument.
- D. The Thread.setPriority() method can take t1 as an argument.
- E. The Object.notify() method arbitrarily chooses which thread to notify.

**Answer: E**

**Question: 31**

**Given that Triangle implements Runnable, and:**

- 31. void go() throws Exception {
- 32. Thread t = new Thread(new Triangle());
- 33. t.start();
- 34. for(int x = 1; x < 100000; x++) {
- 35. //insert code here
- 36. if(x%100 == 0) System.out.print("g");
- 37. }}
- 38. public void run() {
- 39. try {
- 40. for(int x = 1; x < 100000; x++) {

41. // insert the same code here

42. if(x%100 == 0) System.out.print("t");

43. }

44. } catch (Exception e) { }

45. }

**Which two statements, inserted independently at both lines 35 and 41, tend to allow both threads to temporarily pause and allow the other thread to execute? (Choose two.)**

- A. Thread.wait();
- B. Thread.join();
- C. Thread.yield();
- D. Thread.sleep(1);
- E. Thread.notify();

**Answer:** C,D

## 10. Java.lang Package

Q:01 Given:

```
11. public class Person {  
12.     private String name, comment;  
13.     private int age;  
14.     public Person(String n, int a, String c) {  
15.         name = n; age = a; comment = c;  
16.     }  
17.     public boolean equals(Object o) {  
18.         if (! (o instanceof Person)) return false;  
19.         Person p = (Person)o;  
20.         return age == p.age && name.equals(p.name);  
21.     }  
22. }
```

What is the appropriate definition of the hashCode method in class Person?

- A. return super.hashCode();
- B. return name.hashCode() + age \* 7;
- C. return name.hashCode() + comment.hashCode() / 2;
- D. return name.hashCode() + comment.hashCode() / 2 - age \* 3;

Answer: B

Q: 02 Given this method in a class:

```
21. public String toString() {
```



```
22. StringBuffer buffer = new StringBuffer();
23. buffer.append('<');
24. buffer.append(this.name);
25. buffer.append('>');
26. return buffer.toString();
27. }
```

**Which statement is true?**

- A. This code is NOT thread-safe.
- B. The programmer can replace StringBuffer with StringBuilder with no other changes.
- C. This code will perform poorly. For better performance, the code should be rewritten:  
  
return "<" + this.name + ">";
- D. This code will perform well and converting the code to use StringBuilder will not enhance the performance.

**Answer: B**

**Q: 03 Given:**

```
11. public void testIfA() {
12. if (testIfB("True")) {
13. System.out.println("True");
14. } else {
15. System.out.println("Not true");
16. }
17. }
18. public Boolean testIfB(String str) {
19. return Boolean.valueOf(str);
20. }
```

**What is the result when method testIfA is invoked?**

- A. True
- B. Not true
- C. An exception is thrown at runtime.
- D. Compilation fails because of an error at line 12.
- E. Compilation fails because of an error at line 19.

**Answer: A**

**Q: 04 Given:**

```
1. public class Boxer1{  
2. Integer i;  
3. int x;  
4. public Boxer1(int y) {  
5. x = i+y;  
6. System.out.println(x);  
7. }  
8. public static void main(String[] args) {  
9. new Boxer1(new Integer(4));  
10. }  
11. }
```

**What is the result?**

- A. The value "4" is printed at the command line.
- B. Compilation fails because of an error in line 5.
- C. Compilation fails because of an error in line 9.
- D. A NullPointerException occurs at runtime.
- E. A NumberFormatException occurs at runtime.

F. An IllegalStateException occurs at runtime.

**Answer: D**

**Q: 05 Given:**

```
1. public class TestString3 {  
2. public static void main(String[] args) {  
3. // insert code here  
5. System.out.println(s);  
6. }  
7. }
```

**Which two code fragments, inserted independently at line 3, generate the output 4247? (Choose two.)**

- A. String s = "123456789";  
s = (s-"123").replace(1,3,"24") - "89";
- B. StringBuffer s = new StringBuffer("123456789");  
s.delete(0,3).replace(1,3,"24").delete(4,6);
- C. StringBuffer s = new StringBuffer("123456789");  
s.substring(3,6).delete(1,3).insert(1, "24");
- D. StringBuilder s = new StringBuilder("123456789");  
s.substring(3,6).delete(1,2).insert(1, "24");
- E. StringBuilder s = new StringBuilder("123456789");  
s.delete(0,3).delete(1,3).delete(2,5).insert(1, "24");

**Answer: B, E**

**Q: 06 Given:**

```
11. public static void test(String str) {
```

```
12. int check = 4;
13. if (check == str.length()) {
14. System.out.print(str.charAt(check - 1) + ", ");
15. } else {
16. System.out.print(str.charAt(0) + ", ");
17. }
18. }
```

and the invocation:

```
21. test("four");
22. test("tee");
23. test("to");
```

What is the result?

- A. r, t, t,
- B. r, e, o,
- C. Compilation fails.
- D. An exception is thrown at runtime.

**Answer: C**

**Q: 07 Given:**

```
11. public class Person {
12. private String name;
13. public Person(String name) {
14. this.name = name;
15. }
16. public boolean equals(Object o) {
17. if ( ! o instanceof Person ) return false;
```

```
18. Person p = (Person) o;  
19. return p.name.equals(this.name);  
20. }  
21. }
```

**Which statement is true?**

- A. Compilation fails because the hashCode method is not overridden.
- B. A HashSet could contain multiple Person objects with the same name.
- C. All Person objects will have the same hash code because the hashCode method is not overridden.
- D. If a HashSet contains more than one Person object with name="Fred", then removing another Person, also  
with name="Fred", will remove them all.

**Answer: B**

**Q: 08 Which two statements are true about the hashCode method? (Choose two.)**

- A. The hashCode method for a given class can be used to test for object equality and object inequality for that  
class.
- B. The hashCode method is used by the java.util.SortedSet collection class to order the elements within that set.
- C. The hashCode method for a given class can be used to test for object inequality, but NOT object equality, for  
that class.
- D. The only important characteristic of the values returned by a hashCode method is that the distribution of  
values must follow a Gaussian distribution.
- E. The hashCode method is used by the java.util.HashSet collection class to group the elements within that set

into hash buckets for swift retrieval.

**Answer: C, E**

**Q: 09 Click the Task button.**

Place the code into the GenericB class definition to make the class compile successfully.

```
import java.util.*;
```

```
public class GenericB<Place> {
```

```
    public Place foo;
```

```
    public void setFoo(Place foo) {
        this.foo = foo;
    }
```

```
    public Place getFoo() {
        return foo;
    }
```

```
    public static void main (String[] args) {
        GenericB<Cat> bar = new GenericB<Cat>();
        bar.setFoo(new Cat());
        Cat c = bar.getFoo();
    }
}
```

```
interface Pet { }
class Cat implements Pet{ }
```

**Code**

? extends Pet

T extends Pet

? implements Pet

T implements Pet

Fet extends T

?

T

<?>

Pet

Done

1.<T extends Pet>

2. T

3.T

4.T

**Q: 10 Given:**

10. public class MyClass {

11.

12. public Integer startingI;

```
13. public void methodA() {  
14. Integer i = new Integer(25);  
15. startingI = i;  
16. methodB(i);  
17. }  
18. private void methodB(Integer i2) {  
19. i2 = i2.intValue();  
20.  
21. }  
22. }
```

If methodA is invoked, which two are true at line 20? (Choose two.)

- A. i2 == startingI returns true.
- B. i2 == startingI returns false.
- C. i2.equals(startingI) returns true.
- D. i2.equals(startingI) returns false.

**Answer: B, C**

**Question: 11**

**Given:**

```
11. public String makinStrings() {  
12. String s = "Fred";  
13. s = s + "47";  
14. s = s.substring(2, 5);  
15. s = s.toUpperCase();  
16. return s.toString();  
17. }
```

How many String objects will be created when this method is invoked?

- A. 1
- B. 2

- C. 3                      D. 4  
E. 5                      F. 6

**Answer: E**

**Q: 12 Given:**

**22. `StringBuilder sb1 = new StringBuilder("123");`**

**23. `String s1 = "123";`**

**24. `// insert code here`**

**25. `System.out.println(sb1 + " " + s1);`**

**Which code fragment, inserted at line 24, outputs "123abc 123abc"?**

- A. `sb1.append("abc"); s1.append("abc");`  
B. `sb1.append("abc"); s1.concat("abc");`  
C. `sb1.concat("abc"); s1.append("abc");`  
D. `sb1.concat("abc"); s1.concat("abc");`  
E. `sb1.append("abc"); s1 = s1.concat("abc");`  
F. `sb1.concat("abc"); s1 = s1.concat("abc");`  
G. `sb1.append("abc"); s1 = s1 + s1.concat("abc");`  
H. `sb1.concat("abc"); s1 = s1 + s1.concat("abc");`

**Answer: E**

**Q: 13 Given:**

- 1. `public class BuildStuff {`**  
**2. `public static void main(String[] args) {`**  
**3. `Boolean test = new Boolean(true);`**  
**4. `Integer x = 343;`**  
**5. `Integer y = new BuildStuff().go(test, x);`**  
**6. `System.out.println(y);`**



7. }

8. int go(Boolean b, int i) {

9. if(b) return (i/7);

10. return (i/49);

11. }

12. }

What is the result?

A. 7

B. 49

C. 343

D. Compilation fails.

E. An exception is thrown at runtime.

Answer: B

Q: 14 Given:

Given:

1. public class KungFu {

2. public static void main(String[] args) {

3. Integer x = 400;

4. Integer y = x;

5. x++;

6. StringBuilder sb1 = new StringBuilder("123");

7. StringBuilder sb2 = sb1;

8. sb1.append("5");

9. System.out.println((x==y) + " " + (sb1==sb2));

10. }

11. }

**What is the result?**

- A. true true
- B. false true
- C. true false
- D. false false
- E. Compilation fails.
- F. An exception is thrown at runtime.

**Answer:** B

**Q: 15 Given:**

**Which two scenarios are NOT safe to replace a StringBuffer object with a StringBuilder object? (Choose two.)**

- A. When using versions of Java technology earlier than 5.0.
- B. When sharing a StringBuffer among multiple threads.
- C. When using the java.io class StringBufferInputStream.
- D. When you plan to reuse the StringBuffer to build more than one string.

**Answer:** A,B

## 11. java.io package and Serialization

Q: 01 Click the Task button.

Chain these constructors to create objects to read from a file named "in" and to write to a file named "out."

```
reader =   "in" );
writer =    'out' );
```

### Constructors

|                                                         |                                                                                              |                                                                             |
|---------------------------------------------------------|----------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------|
| <input in");"="" type="text" value="new FileReader("/>  | <input out");"="" type="text" value="new PrintWriter("/>                                     | <input out"));"="" type="text" value="new BufferedWriter(new FileWriter("/> |
| <input out");"="" type="text" value="new FileWriter("/> | <input out")));"="" type="text" value="new PrintWriter(new BufferedWriter(new FileWriter("/> | <input out"));"="" type="text" value="new BufferedWriter(new FileWriter("/> |

Solution:

```
reader = new BufferedReader(new FileReader("in"));
writer = new PrintWriter (new BufferedWriter (new FileWriter("out")));
```

Q: 02 Given:

12. import java.io.\*;
13. public class Forest implements Serializable {
14. private Tree tree = new Tree();
15. public static void main(String [] args) {
16. Forest f = new Forest();
17. try {

```

18. FileOutputStream fs = new FileOutputStream("Forest.ser");
19. ObjectOutputStream os = new ObjectOutputStream(fs);
20. os.writeObject(f); os.close();
21. } catch (Exception ex) { ex.printStackTrace(); }
22. }}
23.
24. class Tree { }

```

What is the result?

- A. Compilation fails.
- B. An exception is thrown at runtime.
- C. An instance of Forest is serialized.
- D. An instance of Forest and an instance of Tree are both serialized.

Answer: B

Q: 03 Click the Task button.

Place the code fragments into position to use a BufferedReader to read an entire text file.

```

class PrintFile {
    public static void main(String[] args){
        BufferedReader buffReader = null;
        //more code here to initialize buffReader
        try {
            String temp;
            while(   ) {
                System.out.println(temp);
            }
        } catch (  ) {
            e.printStackTrace();
        }
    }
}

```

**Code Fragments**

|                                 |                        |
|---------------------------------|------------------------|
| (temp = buffReader.readLine()); | ! buffReader.hasNext() |
| (temp = buffReader.nextLine()); | { } if (Exception e) { |
| != null                         | FileNotFoundException  |

Solution:

1. (temp = buffReader.readLine())
2. != null
3. (IOException e){

**Q: 04** Assuming that the `serializeBanana()` and the `deserializeBanana()` methods will correctly use Java serialization and given:

```
13. import java.io.*;
14. class Food implements Serializable {int good = 3;}
15. class Fruit extends Food {int juice = 5;}
16. public class Banana extends Fruit {
17. int yellow = 4;
18. public static void main(String [] args) {
19. Banana b = new Banana(); Banana b2 = new Banana();
20. b.serializeBanana(b); // assume correct serialization
21. b2 = b.deserializeBanana(); // assume correct
22. System.out.println("restore "+b2.yellow+ b2.juice+b2.good);
24. }
25. // more Banana methods go here 50. }
```

**What is the result?**

- A. restore 400                      B. restore 403
- C. restore 453                      D. Compilation fails.
- E. An exception is thrown at runtime.

**Answer: C**

**Q: 05** Which three statements concerning the use of the `java.io.Serializable` interface are true? (Choose three.)

- A. Objects from classes that use aggregation cannot be serialized.
- B. An object serialized on one JVM can be successfully deserialized on a different JVM.
- C. The values in fields with the `volatile` modifier will NOT survive serialization and deserialization.

D. The values in fields with the transient modifier will NOT survive serialization and deserialization.

E. It is legal to serialize an object of a type that has a supertype that does NOT implement java.io.Serializable.

**Answer: B, D, E**

**Q: 06 Assuming that the serializeBanana2() and the deserializeBanana2() methods will correctly use Java serialization and given:**

```
13. import java.io.*;
14. class Food {Food() { System.out.print("1"); }}
15. class Fruit extends Food implements Serializable {
16. Fruit() { System.out.print("2"); }}
17. public class Banana2 extends Fruit { int size = 42;
18. public static void main(String [] args) {
19. Banana2 b = new Banana2();
20. b.serializeBanana2(b); // assume correct serialization
21. b = b.deserializeBanana2(b); // assume correct
22. System.out.println(" restored " + b.size + " "); }
23. // more Banana2 methods
24. }
```

**What is the result?**

- |                       |                                       |
|-----------------------|---------------------------------------|
| A. Compilation fails. | B. 1 restored 42                      |
| C. 12 restored 42     | D. 121 restored 42                    |
| E. 1212 restored 42   | F. An exception is thrown at runtime. |

**Answer: D**

**Q: 7 When comparing java.io.BufferedWriter to java.io.FileWriter, which capability exists as a method in only one of the two?**

- A. closing the stream
- B. flushing the stream
- C. writing to the stream
- D. marking a location in the stream
- E. writing a line separator to the stream

**Answer: E**

**Question: 8**

**Given:**

```
10. class MakeFile {  
11.     public static void main(String[] args) {  
12.         try {  
13.             File directory = new File("d");  
14.             File file = new File(directory,"f");  
15.             if(!file.exists()) {  
16.                 file.createNewFile();  
17.             }  
18.         } catch (IOException e) {  
19.             e.printStackTrace  
20.         }  
21.     }  
22. }
```

**The current directory does NOT contain a directory named "d."**

**Which three are true? (Choose three.)**

- A. Line 16 is never executed.

- B. An exception is thrown at runtime.
- C. Line 13 creates a File object named "d."
- D. Line 14 creates a File object named "f."
- E. Line 13 creates a directory named "d" in the file system.
- F. Line 16 creates a directory named "d" and a file 'f' within it in the file system.
- G. Line 14 creates a file named 'f' inside of the directory named "d" in the file system.

**Answer: BCD**

**Q: 09 Click the Task button.**

The `doesFileExist` method takes an array of directory names representing a path from the root filesystem and a file name. The method returns true if the file exists, false if it does not.

Place the code fragments in position to complete this method.

```
public static boolean doesFileExist(String[] directories, String filename) {
    Place here
    for ( String dir : directories ) {
        Place here
    }
    Place here
    Place here
}
```

#### Code Fragments

|                                                |                                                    |                                                    |
|------------------------------------------------|----------------------------------------------------|----------------------------------------------------|
| <code>path = path.getSubdirectory(dir);</code> | <code>return ! file.isNew();</code>                | <code>return (file != null);</code>                |
| <code>String path = "";</code>                 | <code>path = path.getFile(filename);</code>        | <code>File path = new File("");</code>             |
| <code>return file.exists();</code>             | <code>return path.isFile();</code>                 | <code>File file = new File(path, filename);</code> |
| <code>path = new File(path, dir);</code>       | <code>File path = new File(File.separator);</code> | <code>path = path + File.separator + dir;</code>   |

**Solution:**

1. `String path=""`;



2. path=path+File.separator+dir;
3. File file=new File(path,filename);
4. return file.exists();

**Q:10** Click the Exhibit button.

**Which code, inserted at line 14, will allow this class to correctly serialize and deserialize?**

```
1. import java.io.*;
2. public class Foo implements Serializable
{
3.     public int x, y;
4.     public Foo( int x, int y ) { this.x =
x; this.y = y; }
5.
6.     private void writeObject(
ObjectOutputStream s )
7.         throws IOException {
8.         s.writeInt(x); s.writeInt(y) ;
9.     }
10.
11.    private void readObject(
ObjectInputStream s )
12.        throws IOException,
ClassNotFoundException {
13.
14.        // insert code here
15.
16.    }
17. }
```

- A. s.defaultReadObject();
- B. this = s.defaultReadObject();
- C. y = s.readInt(); x = s.readInt();
- D. x = s.readInt(); y = s.readInt();

**Answer: D**

**Question: 11**

**Given:**

10. public class Foo implements java.io.Serializable {
11. private int x;
12. public int getX() { return x; }
12. public Foo(int x){this.x=x; }
13. private void writeObject( ObjectOutputStream s)
14. throws IOException {
15. // insert code here
16. }
17. }

Which code fragment, inserted at line 15, will allow Foo objects to be correctly serialized and deserialized?

- A. s.writeInt(x);
- B. s.serialize(x);
- C. s.writeObject(x);
- D. s.defaultWriteObject();

**Answer: D**

## 12 Click the Task button.

Place the Fragments into the program, so that the program will get lines from a text file, display them, and then close all the resources.

**Program**

```
import java.io.*;

public class ReadFile {
    public static void main(String [] args) {
        try {
            File f1 = new File("MyText.txt");
             Place here  = new  Place here (x1);
             Place here  x4 = new  Place here (x2);
            String x3 = null;
            while ((x3 =   Place here ()) != null) {
                System.out.println(x3);
            }   Place here ();
        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }
}
```

**Code Fragments**

- BufferedReader
- StreamReader
- FileReader
- readLine
- readIn
- read
- closeFile
- close
- x1  x2
- x3  x4

**Solution:**

```
import java.io.*;

public class ReadFile{

    public static void main(String s[ ]){

        try {

            File x1=new File("MyText.txt");

            FileReader x2=new FileReader(x1);

            BufferedReader x4=new BufferedReader(x2);

            String s3=null;

            while((x3 = x4.readLine()) != null ) {

                System.out.println(x3);

            }x4.close();

        }catch(Exception e){

            e.printStackTrace();

        }

    }

}
```

**Question: 13**

**Which capability exists only in java.io.FileWriter?**

- A. Closing an open stream.
- B. Flushing an open stream.
- C. Writing to an open stream.
- D. Writing a line separator to an open stream.

**Answer: D**

**Question: 14**

Given that the current directory is empty, and that the user has read and write permissions, and the following:

```
11. import java.io.*;
12. public class DOS {
13.     public static void main(String[] args) {
14.         File dir = new File("dir");
15.         dir.mkdir();
16.         File f1 = new File(dir, "f1.txt");
17.         try {
18.             f1.createNewFile();
19.         } catch (IOException e) { ; }
20.         File newDir = new File("newDir");
21.         dir.renameTo(newDir);
22.     }
23. }
```

**Which statement is true?**

- A. Compilation fails.
- B. The file system has a new empty directory named dir.
- C. The file system has a new empty directory named newDir.
- D. The file system has a directory named dir, containing a file f1.txt.
- E. The file system has a directory named newDir, containing a file f1.txt.

**Answer:** E

**Question:** 15

**Given:**

```
1. public class LineUp {
2.     public static void main(String[] args) {
```

3. double d = 12.345;

4. // insert code here

5. }

6. }

Which code fragment, inserted at line 4, produces the output | 12.345|?

A. System.out.printf("|%7d| \n", d);

B. System.out.printf("|%7f| \n", d);

C. System.out.printf("|%3.7d| \n", d);

D. System.out.printf("|%3.7f| \n", d);

E. System.out.printf("|%7.3d| \n", d);

F. System.out.printf("|%7.3f| \n", d);

**Answer:** F

**Question: 16**

**Given:**

5. import java.io.\*;

6. public class Talk {

7. public static void main(String[] args) {

8. Console c = new Console();

9. String pw;

10. System.out.print("password: ");

11. pw = c.readLine();

12. System.out.println("got " + pw);

13. }

14. }

If the user types the password aiko when prompted, what is the result?

A. password:

got

B. password:

got aiko

C. password: aiko

got aiko

D. An exception is thrown at runtime.

E. Compilation fails due to an error on line 8.

**Answer:** E

**Question: 17**

Given that the current directory is empty, and that the user has read and write privileges to the current directory, and the following:

1. `import java.io.*;`

2. `public class Maker {`

3. `public static void main(String[] args) {`

4. `File dir = new File("dir");`

5. `File f = new File(dir, "f");`

6. `}`

7. `}`

**Which statement is true?**

A. Compilation fails.

B. Nothing is added to the file system.

C. Only a new file is created on the file system.

D. Only a new directory is created on the file system.

E. Both a new file and a new directory are created on the file system.

**Answer:** B

**Question: 18**

Given:

12. String csv = "Sue,5,true,3";

13. Scanner scanner = new Scanner( csv );

14. scanner.useDelimiter(", ");

15. int age = scanner.nextInt();

What is the result?

A. Compilation fails.

B. After line 15, the value of age is 5.

C. After line 15, the value of age is 3.

D. An exception is thrown at runtime.

**Answer:** D

**Question: 19**

**Given that c is a reference to a valid java.io.Console object, which two code fragments read a line of text from the console? (Choose two.)**

A. String s = c.readLine();

B. char[] c = c.readLine();

C. String s = c.readConsole();

D. char[] c = c.readConsole();

E. String s = c.readLine("%s", "name ");

F. char[] c = c.readLine("%s", "name ");

**Answer:** A,E

**Question: 20**

Given that `c` is a reference to a valid `java.io.Console` object, and:

11. `String pw = c.readPassword("%s", "pw: ");`

12. `System.out.println("got " + pw);`

13. `String name = c.readLine("%s", "name: ");`

14. `System.out.println(" got ", name);`

If the user types `fido` when prompted for a password, and then responds `bob` when prompted for a name, what is the result?

- A. `pw: got fido name: bob got bob`
- B. `pw: fido got fido name: bob got bob`
- C. `pw: got fido name: bob got bob`
- D. `pw: fido got fido name: bob got bob`
- E. Compilation fails.
- F. An exception is thrown at runtime.

**Answer: E**

**Question: 21**

Given the following six method names:

`addListener`

`addMouseListener`

`setMouseListener`

`deleteMouseListener`

`removeMouseListener`

`registerMouseListener`

How many of these method names follow JavaBean Listener naming rules?

- A. 1
- B. 2
- C. 3



D. 4

E. 5

**Answer: B****Question: 22**

Click the Task button.

**Drag and Drop**

Place the code fragments into position to produce the output:  
true true false

**Code**

```
Scanner scanner = new Scanner( "One,5,true,3,true,6,7,false");
scanner.useDelimiter(",");
while (  ) {
    if (  ) {
        System.out.print(  + " ");
    } else  ;
}
```

**Code Fragments**

**Answer:**

**Drag and Drop**

Place the code fragments into position to produce the output:  
true true false

**Code**

```
Scanner scanner = new Scanner( "One,5,true,3,true,6,7,false");
scanner.useDelimiter(",");
while (  ) {
    if (  ) {
        System.out.print(  + " ");
    } else  ;
}
```

**Code Fragments**

## 12. Collections and Generics

Q: 01 Given:

34. `HashMap props = new HashMap();`

35. `props.put("key45", "some value");`

36. `props.put("key12", "some other value");`

37. `props.put("key39", "yet another value");`

38. `Set s = props.keySet();`

39. `// insert code here`

What, inserted at line 39, will sort the keys in the props HashMap?

A. `Arrays.sort(s);`

B. `s = new TreeSet(s);`

C. `Collections.sort(s);`

D. `s = new SortedSet(s);`

Answer: B

Q: 02 Click the Exhibit button.

Which statement is true about the set variable on line 12?

```
1. import java.util.*;
2. public class TestSet {
3.     enum Example { ONE, TWO, THREE }
4.     public static void main(String[] args)
5.     {
6.         Collection coll = new ArrayList();
7.         coll.add(Example.THREE);
8.         coll.add(Example.THREE);
9.         coll.add(Example.THREE);
10.        coll.add(Example.TWO);
11.        coll.add(Example.TWO);
12.        coll.add(Example.ONE);
13.        Set set = new HashSet(coll);
14.    }
```

- A. The set variable contains all six elements from the coll collection, and the order is guaranteed to be preserved.
- B. The set variable contains only three elements from the coll collection, and the order is guaranteed to be preserved.
- C. The set variable contains all six elements from the coll collection, but the order is NOT guaranteed to be preserved.
- D. The set variable contains only three elements from the coll collection, but the order is NOT guaranteed to be preserved.

**Answer: D**

**Q: 03 Given:**

```
23. Object [] myObjects = {  
24. new Integer(12),  
25. new String("foo"),  
26. new Integer(5),  
27. new Boolean(true)  
28. };  
29. Arrays.sort(myObjects);  
30. for(int i=0; i<myObjects.length; i++) {  
31. System.out.print(myObjects[i].toString());  
32. System.out.print(" ");  
33. }
```

**What is the result?**

- A. Compilation fails due to an error in line 23.
- B. Compilation fails due to an error in line 29.
- C. A ClassCastException occurs in line 29.
- D. A ClassCastException occurs in line 31.
- E. The value of all four objects prints in natural order.

Answer: C

Q: 04 Click the Task button.

Place code into the class so that it compiles and generates the output answer=42. Note: Code options may be used more than once.

### Class

```
public class Place here {  
    private Place here object;  
    public Place here (Place here object) {  
        this.object = object;  
    }  
    public Place here getObject() {  
        return object;  
    }  
  
    public static void main(String[] args) {  
        Gen<String> str = new Gen<String>("answer");  
        Gen<Integer> intg = new Gen<Integer>(42);  
        System.out.println(str.getObject() + "=" +  
            intg.getObject());  
    }  
}
```

### Code Options

Gen<T>

Gen<?>

Gen

?

T

Done

Solution:

1.Gen<T>

2.T

3.Gen

4.T

5.T

Q: 05 Click the Task button.

Given:

```
public void takeList(List<? extends String> list) {
    // insert code here
}
```

Place the Compilation Results on each code statement to indicate whether or not that code will compile if inserted into the takeList() method.

#### Code Statements

`list.add("Foo");`

`list = new ArrayList<String>();`

`list = new ArrayList<Object>();`

`String s = list.get(0);`

`Object o = list;`

#### Compilation Results

Compilation succeeds

Compilation fails

Done

Solution:

1. `list.add("foo");` ----- Compilation fails
2. `list = new ArrayList<String>();` -----Compilation succeeds
3. `list=new ArrayList<Object>();` ---- Compilation fails
4. `String s = list.get(0);` ----- Compilation succeeds
5. `Object o = list;` ----- Compilation succeeds

Q: 06 Given:

1. `public class Person {`
2. `private String name;`
3. `public Person(String name) { this.name = name; }`
4. `public boolean equals(Person p) {`
5. `return p.name.equals(this.name);`
6. `}`

7. }

**Which statement is true?**

- A. The equals method does NOT properly override the Object.equals method.
- B. Compilation fails because the private attribute p.name cannot be accessed in line 5.
- C. To work correctly with hash-based data structures, this class must also implement the hashCode method.
- D. When adding Person objects to a java.util.Set collection, the equals method in line 4 will prevent duplicates.

**Answer: A**

**Q: 07 Given:**

1. `import java.util.*;`
2. `public class Old {`
3. `public static Object get0(List list) {`
4. `return list.get(0);`
5. `}`
6. `}`

**Which three will compile successfully? (Choose three.)**

- A. `Object o = Old.get0(new LinkedList());`
- B. `Object o = Old.get0(new LinkedList<?>());`
- C. `String s = Old.get0(new LinkedList<String>());`
- D. `Object o = Old.get0(new LinkedList<Object>());`
- E. `String s = (String)Old.get0(new LinkedList<String>());`

**Answer: A, D, E**

**Q: 08 Given:**

1. `import java.util.*;`

```
2. public class Example {  
3. public static void main(String[] args) {  
4. // insert code here  
5. set.add(new Integer(2));  
6. set.add(new Integer(1));  
7. System.out.println(set);  
8. }  
9. }
```

Which code, inserted at line 4, guarantees that this program will output [1, 2]?

- A. Set set = new TreeSet();
- B. Set set = new HashSet();
- C. Set set = new SortedSet();
- D. List set = new SortedList();
- E. Set set = new LinkedHashSet();

**Answer: A**

**Q: 09 Given:**

```
11. public static Collection get() {  
12. Collection sorted = new LinkedList();  
13. sorted.add("B"); sorted.add("C"); sorted.add("A");  
14. return sorted;  
15. }  
16. public static void main(String[] args) {  
17. for (Object obj: get()) {  
18. System.out.print(obj + " ");  
19. }  
20. }
```

What is the result?

- A. A, B, C,
- B. B, C, A,
- C. Compilation fails.
- D. The code runs with no output.
- E. An exception is thrown at runtime.

**Answer: B**

**Q:10 given**

**Click the Task button.**

Place the correct description of the compiler output on the code fragments to be inserted at lines 4 and 5. The same compiler output may be used more than once.

```
1. import java.util.*;
2. public class X {
3.     public static void main(String[] args) {
4.         // insert code here
5.         // insert code here
6.     }
7.     public static void foo(List<Object> list) {
8.     } }
```

**Code**

ArrayList<String> x1 = new ArrayList<String>();  
foo(x1);

ArrayList<Object> x2 = new ArrayList<String>();  
foo(x2);

ArrayList<Object> x3 = new ArrayList<Object>();  
foo(x3);

ArrayList x4 = new ArrayList();  
foo(x4);

**Compiler Output**

Compilation succeeds.

Compilation fails due to an error in the first statement.

Compilation of the first statement succeeds, but compilation fails due to an error in the second statement.

Done

**Solution:**



1. Compilation of the first statement succeeds ,but compilation fails due to an error in the second statement.
2. Compilation fails due to an error in the first statement
3. Compilation succeeds
4. Compilation succeeds

Q: 11 Given:

```
11. public static Iterator reverse(List list) {  
12. Collections.reverse(list);  
13. return list.iterator();  
14. }  
  
15. public static void main(String[] args) {  
16. List list = new ArrayList();  
17. list.add("1"); list.add("2"); list.add("3");  
18. for (Object obj: reverse(list))  
19. System.out.print(obj + " , ");  
20. }
```

What is the result?

- A. 3, 2, 1,                                      B. 1, 2, 3,  
C. Compilation fails.                              D. The code runs with no output.  
E. An exception is thrown at runtime.

**Answer: C**

Q: 12 Click the Task button.

Given:

```

1. import java.util.*;
2. class A { }
3. class B extends A { }
4. public class Test {
5.     public static void main(String[] args) {
6.         List<A> listA = new LinkedList<A>();
7.         List<B> listB = new LinkedList<B>();
8.         List<Object> listO = new LinkedList<Object>();
9.         // insert code here
10.    }
11.    public static void m1(List<? extends A> list) { }
12.    public static void m2(List<A> list) { }
13. }

```

Place a result onto each method call to indicate what would happen if the method call were inserted at line 9. Note: Results can be used more than once.

| Method Calls |            | Result                             |
|--------------|------------|------------------------------------|
| m1(listA);   | m2(listA); | Does not compile.                  |
| m1(listB);   | m2(listB); | Compiles and runs without error.   |
| m1(listO);   | m2(listO); | An exception is thrown at runtime. |

=====does not compile=====

1.m1(listO);

2.m2(listB);

3.m2(listO);

=====compiles and runs with out error=====

1.m1(listA);

2.m1(listB);

3.m2(listA);

Q: 13 Given:

1. import java.util.\*;

2. public class PQ {

3. public static void main(String[] args) {

```
4. PriorityQueue<String> pq = new PriorityQueue<String>();
5. pq.add("carrot");
6. pq.add("apple");
7. pq.add("banana");
8. System.out.println(pq.poll() + ":" + pq.peek());
9. }
10. }
```

**What is the result?**

- A. apple:apple
- B. carrot:apple
- C. apple:banana
- D. banana:apple
- E. carrot:carrot
- F. carrot:banana

**Answer: C**

**Q: 14 Given:**

```
1. import java.util.*;
2. public class WrappedString {
3.     private String s;
4.     public WrappedString(String s) { this.s = s; }
5.     public static void main(String[] args) {
6.         HashSet<Object> hs = new HashSet<Object>();
7.         WrappedString ws1 = new WrappedString("aardvark");
8.         WrappedString ws2 = new WrappedString("aardvark");
9.         String s1 = new String("aardvark");
10.        String s2 = new String("aardvark");
```

11. `hs.add(ws1); hs.add(ws2); hs.add(s1); hs.add(s2);`

12. `System.out.println(hs.size()); } }`

What is the result?

- A. 0
- B. 1
- C. 2
- D. 3
- E. 4
- F. Compilation fails.
- G. An exception is thrown at runtime.

**Answer: D**

**Q: 15 Given:**

- 11. `public class Key {`
- 12. `private long id1;`
- 13. `private long id2;`
- 14.
- 15. `// class Key methods`
- 16. `}`

A programmer is developing a class `Key`, that will be used as a key in a standard `java.util.HashMap`. Which two methods should be overridden to assure that `Key` works correctly as a key? (Choose two.)

- A. `public int hashCode()`
- B. `public boolean equals(Key k)`
- C. `public int compareTo(Object o)`
- D. `public boolean equals(Object o)`
- E. `public boolean compareTo(Key k)`

**Answer: A, D**

**Q: 16** Given a pre-generics implementation of a method:

```
11. public static int sum(List list) {  
12.     int sum = 0;  
13.     for ( Iterator iter = list.iterator(); iter.hasNext(); ) {  
14.         int i = ((Integer)iter.next()).intValue();  
15.         sum += i;  
16.     }  
17.     return sum;  
18. }
```

**Which three changes must be made to the method sum to use generics? (Choose three.)**

- A. remove line 14
- B. replace line 14 with "int i = iter.next();"
- C. replace line 13 with "for (int i : intList) {"
- D. replace line 13 with "for (Iterator iter : intList) {"
- E. replace the method declaration with "sum(List<int> intList)"
- F. replace the method declaration with "sum(List<Integer> intList)"

**Answer: A, C, F**

**Q: 17** Given:

```
11. // insert code here  
12. private N min, max;  
13. public N getMin() { return min; }  
14. public N getMax() { return max; }  
15. public void add(N added) {
```

```
16. if (min == null || added.doubleValue() < min.doubleValue()) 17. min = added;
18. if (max == null || added.doubleValue() > max.doubleValue()) 19. max = added;
20. }
21. }
```

**Which two, inserted at line 11, will allow the code to compile? (Choose two.)**

- A. public class MinMax<?> {
- B. public class MinMax<? extends Number> {
- C. public class MinMax<N extends Object> {
- D. public class MinMax<N extends Number> {
- E. public class MinMax<? extends Object> {
- F. public class MinMax<N extends Integer> {

**Answer: D, F**

**Q: 18 Given:**

```
1. import java.util.*;
2.
3. public class LetterASort{
4. public static void main(String[] args) {
5. ArrayList<String> strings = new ArrayList<String>();
6. strings.add("aAaA");
7. strings.add("AaA");
8. strings.add("aAa");
9. strings.add("AAaa");
10. Collections.sort(strings);
11. for (String s : strings) { System.out.print(s + " "); }
12. }
13. }
```



```
private HashMap<String , Integer> map=
    new HashMap<String , Integer >();

public void put(String name , int Value) {
    map.put(name , Value);
}

public Set<String> getNames() {
    return map.keySet();
}
}
```

**Q: 20 Which two statements are true about the hashCode method? (Choose two.)**

- A. The hashCode method for a given class can be used to test for object equality and object inequality for that class.
- B. The hashCode method is used by the java.util.SortedSet collection class to order the elements within that set.
- C. The hashCode method for a given class can be used to test for object inequality, but NOT object equality, for that class.
- D. The only important characteristic of the values returned by a hashCode method is that the distribution of values must follow a Gaussian distribution.
- E. The hashCode method is used by the java.util.HashSet collection class to group the elements within that set into hash buckets for swift retrieval.

**Answer: C, E**

**Q: 21 Click the Task button.**



Place the code in the appropriate places such that this program will always output [1, 2].

```
import java.util.*;

public class MyInt Place here Place here {
    public static void main(String[] args) {
        ArrayList<MyInt> list = new ArrayList<MyInt>();
        list.add(new MyInt(2));
        list.add(new MyInt(1));
        Collections.sort(list);
        System.out.println(list);
    }
    private int i;
    public MyInt(int i) { this.i = i; }
    public String toString() { return Integer.toString(i); }

    Place here int Place here {
        MyInt i2 = (MyInt)o;
        return Place here ;
    }
}
```

#### Code

|                            |                              |          |        |            |
|----------------------------|------------------------------|----------|--------|------------|
| implements                 | extends                      | Sortable | Object | Comparable |
| protected                  | public                       | i - i2.i | i      | i2.i - i   |
| compare(MyInt o, MyInt i2) | compare(Object o, Object i2) |          |        |            |
| sort(Object o)             | sort(MyInt o)                |          |        |            |
| compareTo(MyInt o)         | compareTo(Object o)          |          |        |            |

Done

**Solution:**

1.implements

2.comparable

3.public

4.coompareTo(Object o)

5. i

**Q: 22** A programmer has an algorithm that requires a java.util.List that provides an efficient implementation of add(0, object), but does NOT need to support quick random access.

**What supports these requirements?**

- A. java.util.Queue
- B. java.util.ArrayList
- C. java.util.LinearList

D. java.util.LinkedList

**Answer: D**

**Q: 23 Click the Task button.**

Place each Collection Type on the statement to which it applies.

| Statements                                                      | Collection Types |
|-----------------------------------------------------------------|------------------|
| allows access to elements by their integer index                | java.util.Map    |
| defines the method V get(Object key)                            | java.util.Set    |
| is designed for holding elements prior to processing            | java.util.List   |
| contains no pair of elements e1 and e2, such that e1 equals(e2) | java.util.Queue  |

**Solution:**

(1)-----(3)    (2)-----(1)    (3)-----(4)    (4)------(2)

**Q: 24 Given:**

10. interface A { void x(); }

11. class B implements A { public void x() {} public void y() {} }

12. class C extends B { public void x() {} }

**And:**

20. java.util.List<A> list = new java.util.ArrayList<A>();

21. list.add(new B());

22. list.add(new C());

23. for (A a : list) {

24. a.x();

25. a.y();

26. }

**What is the result?**

- A. The code runs with no output.
- B. An exception is thrown at runtime.
- C. Compilation fails because of an error in line 20.
- D. Compilation fails because of an error in line 21.
- E. Compilation fails because of an error in line 23.
- F. Compilation fails because of an error in line 25.

**Answer: F**

**Q: 25 Click the Task button.**

Place the correct description of the compiler output or the code fragments to be inserted at lines 4 and 5. The same compiler output may be used more than once.

```
1. import java.util.*;
2. public class X {
3.     public static void main(String[] args) {
4.         // insert code here
5.         // insert code here
6.     }
7.     public static void foo(List<Object> list) {
8.     } }
```

**Code**

```
ArrayList<String> x1 = new ArrayList<String>();
foo(x1);
```

```
ArrayList<Object> x2 = new ArrayList<String>();
foo(x2);
```

```
ArrayList<Object> x3 = new ArrayList<Object>();
foo(x3);
```

```
ArrayList x4 = new ArrayList();
foo(x4);
```

**Compiler Output**

Compilation succeeds.

Compilation fails due to an error in the first statement.

Compilation of the first statement succeeds, but compilation fails due to an error in the second statement.

Done

**Solution:**

4. **Compilation of the first statement succeeds ,but compilation fails due to an error in the second statement.**
5. **Compilation fails due to an error in the first statement**
6. **Compilation succeeds**
7. **Compilation succeeds**

Q: 26 Click the Task button.

Given:

```
public void takeList(List<? extends String> list) {  
    // insert code here  
}
```

Place the Compilation Results on each code statement to indicate whether or not that code will compile if inserted into the takeList() method.

**Code Statements**

- 
- 
- 
- 
- 

**Compilation Results**

- 
- 

Solution:

1. list.add("foo"); ----- Compilation fails
2. list = new ArrayList<String>(); -----Compilation succeeds
3. list=new ArrayList<Object>(); ---- Compilation fails
4. String s = list.get(0); ----- Compilation succeeds
5. Object o = list; ----- Compilation succeeds

Q: 27 Given:

1. public class Drink implements Comparable {
2. public String name;
3. public int compareTo(Object o) {

4. return 0;

5. }

6. }

and:

20. Drink one = new Drink();

21. Drink two = new Drink();

22. one.name= "Coffee";

23. two.name= "Tea";

23. TreeSet set = new TreeSet();

24. set.add(one);

25. set.add(two);

A programmer iterates over the TreeSet and prints the name of each Drink object.

What is the result?

- A. Tea
- B. Coffee
- C. Coffee Tea
- D. Compilation fails.
- E. The code runs with no output.
- F. An exception is thrown at runtime.

**Answer: B**

**Q:28 Click the Task button.**

Given the class definitions:

```
class Animal { }
class Dog extends Animal { }
```

and the code:

```
public void go() {
    ArrayList<Dog> aList = new ArrayList<Dog>();
    takeList(aList);
}
// insert definition of the takeList() method here
```

Place the correct Compilation Result on each takeList() method definition to indicate whether or not the go() method would compile given that definition.

#### takeList() Method Definition

`public void takeList(ArrayList list) { }`

`public void takeList(ArrayList<Animal> list) { }`

`public void takeList(ArrayList<? extends Animal> list) { }`

`public void takeList(ArrayList<?> list) { }`

`public void takeList(ArrayList<Object> list) { }`

#### Compilation Result

Compilation succeeds.

Compilation fails.

**Solutions:**

compilation fails:

`Public void takeList(ArrayList<Animal> list) { }`

`Public void takeList(ArrayList<Object> list) { }`

compilation Succeeds

All the remaining



Q: 29 Click the Task button.

Given:

```
1. import java.util.*;
2. public class TestGenericConversion {
3.     public static void main(String[] args) {
4.         List list = new LinkedList();
5.         list.add("one");
6.         list.add("two");
7.         System.out.print(((String)list.get(0)).length());
8.     }
9. }
```

Refactor this class to use generics without changing the code's behavior.

```
1. import java.util.*;
2. public class TestGenericConversion {
3.     public static void main(String[] args) {
4.         Place here
5.         list.add("one");
6.         list.add("two");
7.         Place here
8.     }
9. }
```

Code

|                                               |                                                         |
|-----------------------------------------------|---------------------------------------------------------|
| List list = new LinkedList();                 | System.out.print(list.get(0).length());                 |
| List<String> list = new LinkedList<String>(); | System.out.print(list.get<String>().length());          |
| List<String> list = new LinkedList();         | System.out.print(<String>list.get().length());          |
| List list = new LinkedList<String>;           | System.out.print(((List<String>)list.get(0)).length()); |

Solution:

import java.util.\*;

```
public class TestGenericConversion {
    public static void main(String s[]){
        List<String> list=new LinkedList<String>();
        list.add("one");
        list.add("two");
        System.out.println(list.get(0).length()); }
    }
```

Q: 30 Given:

10. abstract public class Employee {
11. protected abstract double getSalesAmount();
12. public double getCommision() {
13. return getSalesAmount() \* 0.15;



14. }

15. }

16. class Sales extends Employee {

17. // insert method here

18. }

**Which two methods, inserted independently at line 17, correctly complete the Sales class?  
(Choose two.)**

- A. double getSalesAmount() { return 1230.45; }
- B. public double getSalesAmount() { return 1230.45; }
- C. private double getSalesAmount() { return 1230.45; }
- D. protected double getSalesAmount() { return 1230.45; }

**Answer: B, D**

**Q: 31 Given:**

```
13. public static void search(List<String> list) {  
14. list.clear();  
15. list.add("b");  
16. list.add("a");  
17. list.add("c");  
18. System.out.println(Collections.binarySearch(list, "a"));  
19. }
```

**What is the result of calling search with a valid List implementation?**

- A. 0
- B. 1
- C. 2
- D. a
- E. b
- F. c
- G. The result is undefined.

**Answer: G**

Q: 32 Given:

```
11. public static void append(List list) { list.add("0042"); }  
12. public static void main(String[] args) {  
13. List<Integer> intList = new ArrayList<Integer>();  
14. append(intList);  
15. System.out.println(intList.get(0));  
16. }
```

What is the result?

- A. 42
- B. 0042
- C. An exception is thrown at runtime.
- D. Compilation fails because of an error in line 13.
- E. Compilation fails because of an error in line 14.

**Answer: B**

Q: 33 Given:

```
11. public class Person {  
12. private name;  
13. public Person(String name) {  
14. this.name = name;  
15. }  
16. public int hashCode() {  
17. return 420;  
18. }  
19. }
```

Which statement is true?

- A. The time to find the value from HashMap with a Person key depends on the size of the map.
- B. Deleting a Person key from a HashMap will delete all map entries for all keys of type Person.
- C. Inserting a second Person object into a HashSet will cause the first Person object to be removed as a duplicate.
- D. The time to determine whether a Person object is contained in a HashSet is constant and does NOT depend on the size of the map.

**Answer: A**

**Q: 34**

**A programmer must create a generic class MinMax and the type parameter of MinMax must implement Comparable. Which implementation of MinMax will compile?**

- A. 

```
class MinMax<E extends Comparable<E>> {  
  
    E min = null;  
  
    E max = null;  
  
    public MinMax() {}  
  
    public void put(E value) { /* store min or max */ }
```
- B. 

```
class MinMax<E implements Comparable<E>> {  
  
    E min = null;  
  
    E max = null;  
  
    public MinMax() {}  
  
    public void put(E value) { /* store min or max */ }
```
- C. 

```
class MinMax<E extends Comparable<E>> {  
  
    <E> E min = null;  
  
    <E> E max = null;  
  
    public MinMax() {}  
  
    public <E> void put(E value) { /* store min or max */ }
```
- D. 

```
class MinMax<E implements Comparable<E>> {  
  
    <E> E min = null;
```

```
<E> E max = null;

public MinMax() {}

public <E> void put(E value) { /* store min or max */ }
```

**Answer: A**

**Q: 35 Given:**

**int[] myArray = new int[] {1, 2, 3, 4, 5}; What allows you to create a list from this array?**

- A. List myList = myArray.asList();
- B. List myList = Arrays.asList(myArray);
- C. List myList = new ArrayList(myArray);
- D. List myList = Collections.fromArray(myArray);

**Answer: B**

**Question: 36**

**Given:**

1. public class Score implements Comparable<Score> {
2. private int wins, losses;
3. public Score(int w, int l) { wins = w; losses = l; }
4. public int getWins() { return wins; }
5. public int getLosses() { return losses; }
6. public String toString() {
7. return "<" + wins + ", " + losses + ">";
8. }
9. // insert code here
10. }

**Which method will complete this class?**

- A. public int compareTo(Object o) { /\*mode code here\*/ }

- B. `public int compareTo(Score other) { /*more code here*/ }`
- C. `public int compare(Score s1,Score s2){ /*more code here*/ }`
- D. `public int compare(Object o1,Object o2){ /*more code here*/ }`

**Answer: B**

**Question: 37**

Click the Exhibit button.

1. `import java.util.*;`
2. `class KeyMaster {`
3. `public int i;`
4. `public KeyMaster(int i) { this.i = i; }`
5. `public boolean equals(Object o) { return i == ((KeyMaster)o).i; }`
6. `public int hashCode() { return i; }`
7. `}`
8. `public class MapIt {`
9. `public static void main(String[] args) {`
10. `Set<KeyMaster> set = new HashSet<KeyMaster>();`
11. `KeyMaster k1 = new KeyMaster(1);`
12. `KeyMaster k2 = new KeyMaster(2);`
13. `set.add(k1); set.add(k1);`
14. `set.add(k2); set.add(k2);`
15. `System.out.print(set.size() + ":");`
16. `k2.i = 1;`
17. `System.out.print(set.size() + ":");`
18. `set.remove(k1);`
19. `System.out.print(set.size() + ":");`
20. `set.remove(k2);`

21. `System.out.print(set.size());`

22. `}`

23. `}`

What is the result?

A. 4:4:2:2

B. 4:4:3:2

C. 2:2:1:0

D. 2:2:0:0

E. 2:1:0:0

F. 2:2:1:1

G. 4:3:2:1

Answer: F

Question: 38

Given:

1. `import java.util.*;`

2. `public class Test {`

3. `public static void main(String[] args) {`

4. `List<String> strings = new ArrayList<String>();`

5. `// insert code here`

6. `}`

7. `}`

Which four, inserted at line 5, will allow compilation to succeed?

(Choose four.)

A. `String s = strings.get(0);`

B. `Iterator i1 = strings.iterator();`

C. `String[] array1 = strings.toArray();`

D. `Iterator<String> i2 = strings.iterator();`

E. `String[] array2 = strings.toArray(new String[1]);`

F. `Iterator<String> i3 = strings.iterator<String>();`

**Answer: ABDE**

**Question: 39**

**Given:**

**class A {}**

**class B extends A {}**

**class C extends A {}**

**class D extends B {}**

**Which three statements are true? (Choose three.)**

- A. The type List<A> is assignable to List.
- B. The type List<B> is assignable to List<A>.
- C. The type List<Object> is assignable to List<?>.
- D. The type List<D> is assignable to List<? extends B>.
- E. The type List<? extends A> is assignable to List<A>.
- F. The type List<Object> is assignable to any List reference.
- G. The type List<? extends B> is assignable to List<? extends A>.

**Answer: CDG**

**Question: 40**

**Given:**

```
11. public void addStrings(List list) {  
12. list.add("foo");  
13. list.add("bar");  
14. }
```

**What must you change in this method to compile without warnings?**

- A. add this code after line 11:

```
list = (List<String>) list;
```

B. change lines 12 and 13 to:

```
list.add<String>("foo");
```

```
list.add<String>("bar");
```

C. change the method signature on line 11 to:

```
public void addStrings(List<? extends String> list) {
```

D. change the method signature on line 11 to:

```
public void addStrings(List<? super String> list) {
```

E. No changes are necessary. This method compiles without warnings.

**Answer: D**

**Question: 41**

**Given:**

```
1. public class Test {  
2. public <T extends Comparable> T findLarger(T x, T y) {  
3. if(x.compareTo(y) > 0) {  
4. return x;  
5. } else {  
6. return y;  
7. }  
8. }  
9. }
```

**and:**

```
22. Test t = new Test();
```

```
23. // insert code here
```

**Which two will compile without errors when inserted at line 23?**



**(Choose two.)**

- A. Object x = t.findLarger(123, "456");
- B. int x = t.findLarger(123, new Double(456));
- C. int x = t.findLarger(123, new Integer(456));
- D. int x = (int) t.findLarger(new Double(123), new Double(456));

**Answer: AC**

**Question: 42**

**Given:**

11. List list = // more code here

12. Collections.sort(list, new MyComparator());

**Which code will sort this list in the opposite order of the sort in line**

**12?**

- A. Collections.reverseSort(list, new MyComparator());
- B. Collections.sort(list, new MyComparator());  
list.reverse();
- C. Collections.sort(list, new InverseComparator(  
new MyComparator()));
- D. Collections.sort(list, Collections.reverseOrder(  
new MyComparator()));

**Answer: D**

**Question: 43**

**Given:**

**ArrayList a = new ArrayList();**

containing the values {"1", "2", "3", "4", "5", "6", "7", "8"}

**Which code will return 2?**

A. Collections.sort(a, a.reverse());

int result = Collections.binarySearch(a, "6");

B. Comparator c = Collections.reverseOrder();

Collections.sort(a, c);

int result = Collections.binarySearch(a, "6");

C. Comparator c = Collections.reverseOrder();

Collections.sort(a, c);

int result = Collections.binarySearch(a, "6",c);

D. Comparator c = Collections.reverseOrder(a);

Collections.sort(a, c);

int result = Collections.binarySearch(a, "6",c);

E. Comparator c = new InverseComparator(new Comparator());

Collections.sort(a);

int result = Collections.binarySearch(a, "6",c);

**Answer: C**

**Question: 44**

**Given:**

11. public class Counter {

12. public static void main(String[] args) {

13. int numArgs = /\* insert code here \*/;

14. }

15. }

and the command line:

java Counter one fred 42

Which code, inserted at line 13, captures the number of arguments passed into the program?

- A. args.count
- B. args.length
- C. args.count()
- D. args.length()
- E. args.getLength()

**Answer: B**

**Question: 45**

**Given:**

- 3. import java.util.\*;
- 4. public class Mapit {
- 5. public static void main(String[] args) {
- 6. Set<Integer> set = new HashSet<Integer>();
- 7. Integer i1 = 45;
- 8. Integer i2 = 46;
- 9. set.add(i1);
- 10. set.add(i1);
- 11. set.add(i2); System.out.print(set.size() + " ");
- 12. set.remove(i1); System.out.print(set.size() + " ");
- 13. i2 = 47;
- 14. set.remove(i2); System.out.print(set.size() + " ");
- 15. }

16. }

What is the result?

- A. 2 1 0
- B. 2 1 1
- C. 3 2 1
- D. 3 2 2
- E. Compilation fails.
- F. An exception is thrown at runtime.

**Answer:** B

**Question: 46**

**Given:**

```
12. import java.util.*;
13. public class Explorer1 {
14.     public static void main(String[] args) {
15.         TreeSet<Integer> s = new TreeSet<Integer>();
16.         TreeSet<Integer> subs = new TreeSet<Integer>();
17.         for(int i = 606; i < 613; i++) 18. if(i%2 == 0) s.add(i);
19.         subs = (TreeSet)s.subSet(608, true, 611, true);
20.         s.add(609);
21.         System.out.println(s + " " + subs);
22.     }
23. }
```

What is the result?

- A. Compilation fails.
- B. An exception is thrown at runtime.
- C. [608, 609, 610, 612] [608, 610]

D. [608, 609, 610, 612] [608, 609, 610]

E. [606, 608, 609, 610, 612] [608, 610]

F. [606, 608, 609, 610, 612] [608, 609, 610]

**Answer:** F

**Question: 47**

**Given:**

3. `import java.util.*;`

4. `public class Quest {`

5. `public static void main(String[] args) {`

6. `String[] colors = {"blue", "red", "green", "yellow", "orange"};`

7. `Arrays.sort(colors);`

8. `int s2 = Arrays.binarySearch(colors, "orange");`

9. `int s3 = Arrays.binarySearch(colors, "violet");`

10. `System.out.println(s2 + " " + s3);`

11. `}`

12. `}`

**What is the result?**

A. 2 -1

B. 2 -4

C. 2 -5

D. 3 -1

E. 3 -4

F. 3 -5

G. Compilation fails.

H. An exception is thrown at runtime.

**Answer:** C

**Question: 48**

**Given:**

```
5. import java.util.*;
6. public class SortOf {
7.     public static void main(String[] args) {
8.         ArrayList<Integer> a = new ArrayList<Integer>();
9.         a.add(1); a.add(5); a.add(3);
11.        Collections.sort(a);
12.        a.add(2);
13.        Collections.reverse(a);
14.        System.out.println(a);
15.    }
16. }
```

**What is the result?**

- A. [1, 2, 3, 5]
- B. [2, 1, 3, 5]
- C. [2, 5, 3, 1]
- D. [5, 3, 2, 1]
- E. [1, 3, 5, 2]
- F. Compilation fails.
- G. An exception is thrown at runtime.

**Answer: C**

**Question: 49**

**Given:**

```
3. import java.util.*;
```

```
4. public class Hancock {  
5. // insert code here 6. list.add("foo");  
7. }  
8. }
```

**Which two code fragments, inserted independently at line 5, will compile without warnings? (Choose two.)**

- A. `public void addStrings(List list) {`
- B. `public void addStrings(List<String> list) {`
- C. `public void addStrings(List<? super String> list) {`
- D. `public void addStrings(List<? extends String> list) {`

**Answer:** B,C

**Question: 50**

**Given a class whose instances, when found in a collection of objects, are sorted by using the `compareTo()` method, which two statements are true? (Choose two.)**

- A. The class implements `java.lang.Comparable`.
- B. The class implements `java.util.Comparator`.
- C. The interface used to implement sorting allows this class to define only one sort sequence.
- D. The interface used to implement sorting allows this class to define many different sort sequences.

**Answer:** A,C

## 13.Inner Classes

Q: 01 Given:

```
10. class Line {  
11.     public static class Point {}  
12. }  
13.  
14. class Triangle {  
15.     // insert code here  
16. }
```

Which code, inserted at line 15, creates an instance of the Point class defined in Line?

- A. Point p = new Point();
- B. Line.Point p = new Line.Point();
- C. The Point class cannot be instantiated at line 15.
- D. Line l = new Line() ; l.Point p = new l.Point();

Answer: B

Q: 02 Given:

```
11. static class A {  
12.     void process() throws Exception { throw new Exception(); }  
13. }  
14. static class B extends A {  
15.     void process() { System.out.println("B "); }  
16. }  
17. public static void main(String[] args) {  
18.     A a = new B();  
19.     a.process();
```



20. }

**What is the result?**

- A. B
- B. The code runs with no output.
- C. An exception is thrown at runtime
- D. Compilation fails because of an error in line 15.
- E. Compilation fails because of an error in line 18.
- F. Compilation fails because of an error in line 19.

**Answer: F**

**Q: 03 Given:**

- 1. package geometry;
- 2. public class Hypotenuse {
- 3. public InnerTriangle it = new InnerTriangle();
- 4. class InnerTriangle {
- 5. public int base;
- 6. public int height;
- 7. }
- 8. }

**Which statement is true about the class of an object that can reference the variable base?**

- A. It can be any class.
- B. No class has access to base.
- C. The class must belong to the geometry package.
- D. The class must be a subclass of the class Hypotenuse.

**Answer: C**

Q: 04 Given:

```
10. class Line {  
11. public class Point { public int x,y;}  
12. public Point getPoint() { return new Point(); }  
13. }  
14. class Triangle {  
15. public Triangle() {  
16. // insert code here  
17. }  
18. }
```

Which code, inserted at line 16, correctly retrieves a local instance of a Point object?

- A. Point p = Line.getPoint();
- B. Line.Point p = Line.getPoint();
- C. Point p = (new Line()).getPoint();
- D. Line.Point p = (new Line()).getPoint();

**Answer: D**

## 14.Internationalization

**Q: 01 Given:**

**11. String test = "This is a test";**

**12. String[] tokens = test.split("\\s");**

**13. System.out.println(tokens.length);**

**What is the result?**

A. 0

B. 1

C. 4

D. Compilation fails.

E. An exception is thrown at runtime

**Answer: D**

**Q: 02 Given:**

**12. System.out.format("Pi is approximately %d.", Math.PI);**

**What is the result?**

A. Compilation fails.

B. Pi is approximately 3.

C. Pi is approximately 3.141593.

D. An exception is thrown at runtime.

**Answer: D**

**Q: 03 Given:**

**33. Date d = new Date(0);**

**34. String ds = "December 15, 2004";**

**35. // insert code here**

**36. try {**

37. `d = df.parse(ds);`

38. `}`

39. `catch(ParseException e) {`

40. `System.out.println("Unable to parse " + ds);`

41. `}`

42. `// insert code here too`

**What creates the appropriate DateFormat object and adds a day to the Date object?**

A. 35. `DateFormat df = DateFormat.getDateFormat();`

42. `d.setTime( (60 * 60 * 24) + d.getTime());`

B. 35. `DateFormat df = DateFormat.getDateInstance();`

42. `d.setTime( (1000 * 60 * 60 * 24) + d.getTime());`

C. 35. `DateFormat df = DateFormat.getDateFormat();`

42. `d.setLocalTime( (1000*60*60*24) + d.getLocalTime());`

D. 35. `DateFormat df = DateFormat.getDateInstance();`

42. `d.setLocalTime( (60 * 60 * 24) + d.getLocalTime());`

**Answer: B**

**Q: 04 Given:**

12. `NumberFormat nf = NumberFormat.getInstance();`

13. `nf.setMaximumFractionDigits(4);`

14. `nf.setMinimumFractionDigits(2);`

15. `String a = nf.format(3.1415926);`

16. `String b = nf.format(2);`

**Which two statements are true about the result if the default locale is Locale.US? (Choose two.)**

A. The value of b is 2.

B. The value of a is 3.14.

C. The value of b is 2.00.

- D. The value of a is 3.141.
- E. The value of a is 3.1415.
- F. The value of a is 3.1416.
- G. The value of b is 2.0000.

**Answer: C, F**

**Q: 05 Click the Task button.**

Given:

```
System.out.printf("Pi is approximately %f and E is approximately %b",  
                  Math.PI, Math.E);
```

Place the values where they would appear in the output.

Pi is approximately

and E is approximately

**Values**

|                                |                                       |                                    |                                      |
|--------------------------------|---------------------------------------|------------------------------------|--------------------------------------|
| <input type="text" value="3"/> | <input type="text" value="3.141593"/> | <input type="text" value="true"/>  | <input type="text" value="Math.PI"/> |
| <input type="text" value="2"/> | <input type="text" value="2.718282"/> | <input type="text" value="false"/> | <input type="text" value="Math.E"/>  |

**Solution:**

**Pi is Approximately 3.141593**

**and E is Approximately true**

**Q: 05 Given:**

- 12. `Date date = new Date();`
- 13. `df.setLocale(Locale.ITALY);`
- 14. `String s = df.format(date);`

The variable df is an object of type `DateFormat` that has been initialized in line 11.

What is the result if this code is run on December 14, 2000?

- A. The value of s is 14-dic-2004.
- B. The value of s is Dec 14, 2000.
- C. An exception is thrown at runtime.
- D. Compilation fails because of an error in line 13.

**Answer: D**

**Q: 06 Given:**

**d is a valid, non-null Date object**

**df is a valid, non-null DateFormat object set to the current locale**

**What outputs the current locale's country name and the appropriate version of d's date?**

- A. 

```
Locale loc = Locale.getLocale();  
System.out.println(loc.getDisplayCountry()  
+ " " + df.format(d));
```
- B. 

```
Locale loc = Locale.getDefault();  
System.out.println(loc.getDisplayCountry()  
+ " " + df.format(d));
```
- C. 

```
Locale loc = Locale.getLocale();  
System.out.println(loc.getDisplayCountry()  
+ " " + df.setDateFormat(d));
```
- D. 

```
Locale loc = Locale.getDefault();  
System.out.println(loc.getDisplayCountry()  
+ " " + df.setDateFormat(d));
```

**Answer: B**

Q: 07

Given a valid DateFormat object named df, and

16. Date d = new Date(0L);

17. String ds = "December 15, 2004";

18. // insert code here

What updates d's value with the date represented by ds?

A. 18. d = df.parse(ds);

B. 18. d = df.getDate(ds);

C. 18. try {

19. d = df.parse(ds);

20. } catch(ParseException e) { };

D. 18. try {

19. d = df.getDate(ds);

20. } catch(ParseException e) { };

Answer: C

Q: 08 Click the Task button.

Place the code fragments into position to produce the output:

true true false

**Code**

```
Scanner scanner = new Scanner( "One,5,true,3,true,6,7,false");  
scanner.useDelimiter(",");
```

```
while (  ) {  
    if (  ) {  
        System.out.print(  + " " );  
    } else  ;  
}
```

**Code Fragments**

Done

Solution:

```
Scanner scanner =new Scanner(One, 5 true,3,true,6,7,false");
scanner.useDelimiter(",");
while( scanner.hasNext()) {
    if ( scanner.hasNextBoolean()){
        system.out.print(scanner.nextBoolean()+" ");
    }else scanner.Next();
}
```

Q: 09 Given:

11. double input = 314159.26;

12. NumberFormat nf = NumberFormat.getInstance(Locale.ITALIAN);

13. String b;

14. //insert code here

Which code, inserted at line 14, sets the value of b to 314.159,26?

A. b = nf.parse( input );

B. b = nf.format( input );

C. b = nf.equals( input );

D. b = nf.parseObject( input );

Answer: B

Q: 10 Given:

12. String csv = "Sue,5,true,3";

13. Scanner scanner = new Scanner( csv );

14. scanner.useDelimiter(",");

15. int age = scanner.nextInt();

What is the result?



- A. Compilation fails.
- B. After line 15, the value of age is 5.
- C. After line 15, the value of age is 3.
- D. An exception is thrown at runtime.

**Answer: D**

**Q: 11 Given:**

- 11. `String test = "a1b2c3";`
- 12. `String[] tokens = test.split("\\d");`
- 13. `for(String s: tokens) System.out.print(s + " ");`

**What is the result?**

- A. a b c
- B. 1 2 3
- C. a1b2c3
- D. a1 b2 c3
- E. Compilation fails.
- F. The code runs with no output.
- G. An exception is thrown at runtime.

**Answer: A**

**Question: 12**

**Given:**

- 14. `DateFormat df;`
- 15. `Date date = new Date();`
- 16. `//insert code here`
- 17. `String s = df.format( date);`

Which two, inserted independently at line 16, allow the code to compile? (Choose two.)

- A. `df= new DateFormat();`
- B. `df= Date.getFormatter();`
- C. `df= date.getFormatter();`
- D. `df= date.getDateFormatter();`
- E. `df= Date.getDateFormatter();`
- F. `df= DateFormat.getInstance();`
- G. `df = DateFormat.getDateInstance();`

**Answer: FG**

**Question: 13**

**Given:**

- 11. `String test = "Test A. Test B. Test C.";`
- 12. `// insert code here`
- 13. `String[] result = test.split(regex);`

Which regular expression inserted at line 12 will correctly split test into "Test A," "Test B," and "Test C"?

- A. `String regex = "";`
- B. `String regex = " ";`
- C. `String regex = ". *";`
- D. `String regex = "\\s";`
- E. `String regex = "\\s*";`
- F. `String regex = "\\w[\\.]+";`

**Answer: E**

**Question: 14**

Given:

5. `import java.util.Date;`

6. `import java.text.DateFormat;`

21. `DateFormat df;`

22. `Date date = new Date();`

23. `// insert code here`

24. `String s = df.format(date);`

Which code fragment, inserted at line 23, allows the code to compile?

A. `df = new DateFormat();`

B. `df = Date.getFormat();`

C. `df = date.getFormat();`

D. `df = DateFormat.getFormat();`

E. `df = DateFormat.getInstance();`

**Answer:** E

## 15.Development

Q: 01 Click the Exhibit button.

Given the fully-qualified class names:

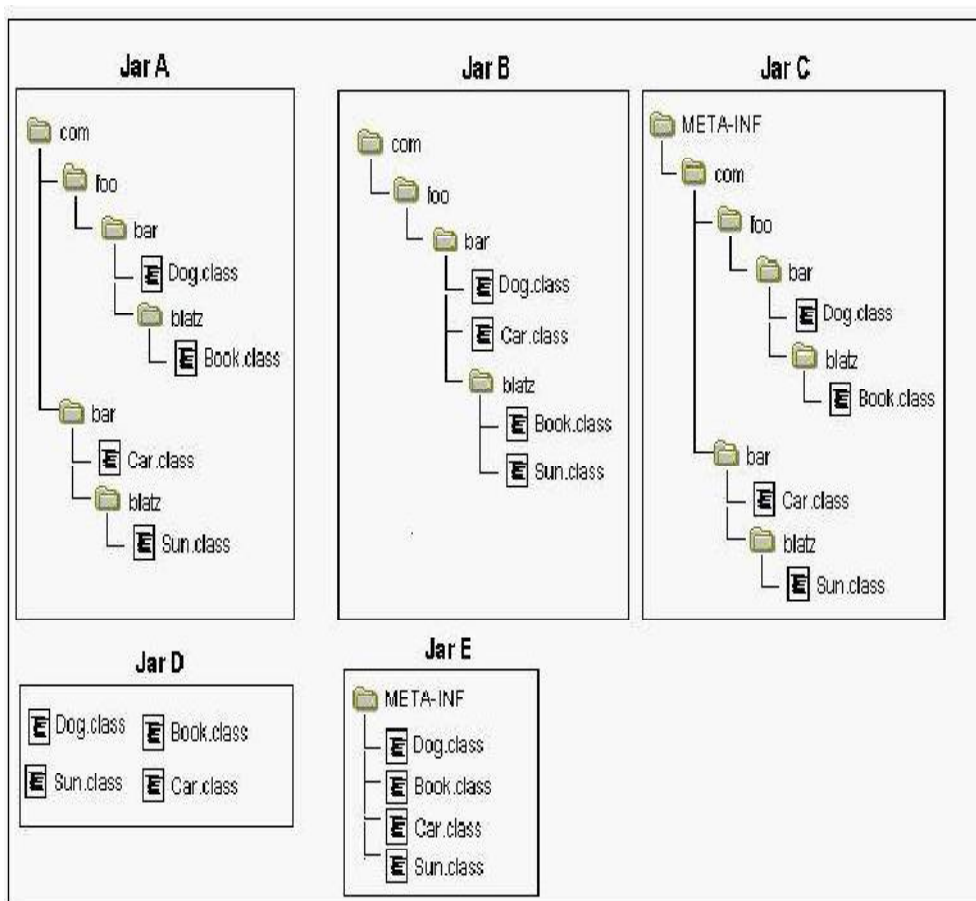
`com.foo.bar.Dog`

`com.foo.bar.blatz.Book`

`com.bar.Car`

`com.bar.blatz.Sun`

Which graph represents the correct directory structure for a JAR file from which those classes can be used by the compiler and JVM?



A. Jar A

B. Jar B

C. Jar C

D. Jar D

E. Jar E

**Answer: A**

**Q: 02**

**A class `games.cards.Poker` is correctly defined in the jar file `Poker.jar`. A user wants to execute the main method of `Poker` on a UNIX system using the command:**  
**`java games.cards.Poker`**What allows the user to do this?

- A. put `Poker.jar` in directory `/stuff/java`, and set the `CLASSPATH` to include `/stuff/java`
- B. put `Poker.jar` in directory `/stuff/java`, and set the `CLASSPATH` to include `/stuff/java/*.jar`
- C. Put `Poker.jar` in directory `/stuff/java`, and set the `CLASSPATH` to include `/stuff/java/Poker.jar`
- D. put `Poker.jar` in directory `/stuff/java/games/cards`, and set the `CLASSPATH` to include `/stuff/java`
- E. put `Poker.jar` in directory `/stuff/java/games/cards`, and set the `CLASSPATH` to include `/stuff/java/*.jar`
- F. put `Poker.jar` in directory `/stuff/java/games/cards`, and set the `CLASSPATH` to include `/stuff/java/Poker.jar`

**Answer: C**

**Q: 03 Given:**

- 11. `public class Commander {`
- 12. `public static void main(String[] args) {`
- 13. `String myProp = /* insert code here */`
- 14. `System.out.println(myProp);`
- 15. `}`
- 16. `}`

**and the command line:**

**`java -Dprop.custom=gobstopper Commander`**

**Which two, placed on line 13, will produce the output gobstopper? (Choose two.)**

- A. `System.load("prop.custom");`
- B. `System.getenv("prop.custom");`
- C. `System.property("prop.custom");`
- D. `System.getProperty("prop.custom");`
- E. `System.getProperties().getProperty("prop.custom");`

**Answer: D, E**

**Q: 04**

**A developer is creating a class Book, that needs to access class Paper. The**

**Paper class is deployed in a JAR named myLib.jar. Which three, taken independently, will allow the developer to use the Paper class while compiling the Book class? (Choose three.)**

- A. The JAR file is located at `$JAVA_HOME/jre/classes/myLib.jar`.
- B. The JAR file is located at `$JAVA_HOME/jre/lib/ext/myLib.jar..`
- C. The JAR file is located at `/foo/myLib.jar` and a classpath environment variable is set that includes `/foo/myLib.jar/Paper.class`.
- D. The JAR file is located at `/foo/myLib.jar` and a classpath environment variable is set that includes `/foo/myLib.jar`.
- E. The JAR file is located at `/foo/myLib.jar` and the Book class is compiled using `javac -cp /foo/myLib.jar/Paper Book.java`.
- F. The JAR file is located at `/foo/myLib.jar` and the Book class is compiled using `javac -d /foo/myLib.jar`  
`Book.java`
- G. The JAR file is located at `/foo/myLib.jar` and the Book class is compiled using `javac -classpath /foo/myLib.jar Book.java`

**Answer: B, D, G**

**Q: 05 Click the Task button.**

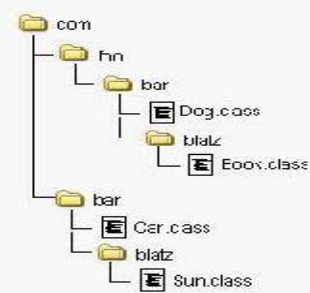
The image at right represents a complete package structure for a set of classes: "com" is the beginning of the fully qualified package name for all classes. Given this package structure, insert the code needed to make the Car class compile and run successfully. At three placeholders must be filled. If fewer than three statements are needed, use the "// blank" option.

place here  
Place here  
Place here

```
public class Car {
    Book book;
    Dog dog;
}
```

```
import com.foo.bar.blatz.*;
import com.bar.*;
package com.bar;
import com.foo.*;
import com.foo.bar.*;
```

```
package com.foo.bar.blatz;
import com.*;
package com;
// blank
import com.foo.bar.Book;
```



Done

**Solution:**

1. package com.bar;
2. import com.foo.bar.\*;
3. import com.foo.bar.blatz.\*;

**Q: 06 Given:**

1. package com.company.application;
- 2.
3. public class MainClass {
4. public static void main(String[] args) {}
5. }

And MainClass exists in the /apps/com/company/application directory. Assume the CLASSPATH environment variable is set to "." (current directory).

**Which two java commands entered at the command line will run MainClass? (Choose two.)**

- A. java MainClass if run from the /apps directory
- B. java com.company.application.MainClass if run from the /apps directory
- C. java -classpath /apps com.company.application.MainClass if run from any directory

D. `java -classpath . MainClass` if run from the `/apps/com/company/application` directory

E. `java -classpath /apps/com/company/application:. MainClass` if run from the `/apps` directory

F. `java com.company.application.MainClass` if run from the `/apps/com/company/application` directory

**Answer: B, C**

**Q: 07** Given a correctly compiled class whose source code is:

1. `package com.sun.sjcp;`

2. `public class Commander {`

3. `public static void main(String[] args) {`

4. `// more code here`

5. `}`

6. `}`

Assume that the class file is located in `/foo/com/sun/sjcp/`, the current directory is `/foo/`, and that the classpath contains `"."` (current directory).

Which command line correctly runs `Commander`?

A. `java Commander`

B. `java com.sun.sjcp.Commander`

C. `java com/sun/sjcp/Commander`

D. `java -cp com.sun.sjcp Commander`

E. `java -cp com/sun/sjcp Commander`

**Answer: B**

**Q: 08**

A UNIX user named Bob wants to replace his chess program with a new one,

but he is not sure where the old one is installed. Bob is currently able to run a Java chess program starting from his home directory `/home/bob` using the command:

`java -classpath /test:/home/bob/downloads/*.jar games.`



Chess Bob's CLASSPATH is set (at login time) to:

`/usr/lib:/home/bob/classes:/opt/java/lib:/opt/java/lib/*.jar`

What is a possible location for the Chess.class file?

- A. /test/Chess.class
- B. /home/bob/Chess.class
- C. /test/games/Chess.class
- D. /usr/lib/games/Chess.class
- E. /home/bob/games/Chess.class
- F. inside jarfile /opt/java/lib/Games.jar (with a correct manifest)
- G. inside jarfile /home/bob/downloads/Games.jar (with a correct manifest)

**Answer: C**

**Q: 09**

Given the following directory structure:

**bigProject**

| --source

| | --Utils.java

|

| --classes

| --

And the following command line invocation:

`javac -d classes source/Utils.java`

Assume the current directory is bigProject, what is the result?

- A. If the compile is successful, Utils.class is added to the source directory.
- B. The compiler returns an invalid flag error.
- C. If the compile is successful, Utils.class is added to the classes directory.
- D. If the compile is successful, Utils.class is added to the bigProject directory.

**Answer: C**

## ***DURGA'S INTERVIEW QUESTIONS***

### **1.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. The synchronized keyword tells the JVM that the method requires a lock in order to run. The JVM then creates the lock and manages the allocation of the lock to threads during execution. To execute static synchronized area compulsory the thread has to get the class level lock.

### **2.Can main method be declared final?**

Yes, the main method can be declared final, in addition to being public static. In addition to final we can declare main method with synchronized modifier also. The following is valid main method declaration

### **3. Can a public class MyClass be defined in a source file named YourClass.java?**

If java source file contains a public class then the name of the source file and name of the public class must be matched. Otherwise compilation fails.

### **4.What is the default value of the local variables?**

The local variables are not initialized to any default value, neither primitives nor object references. If you try to use these variables without initializing them explicitly, the java compiler will not compile the code. It will complain about the local variable not being initialized. Hence,

**Before using local variable it must be initialized**

### **5. Is Empty.java file a valid source file?**

Yes, an empty .java file is a perfectly valid source file.

### **6. What are the different scopes for Java variables?**

The scope of a Java variable is determined by the context in which the variable is declared. Thus a java variable can have one of the three scopes at any given point in time.

**1. Instance :** - These are typical object level variables, they are initialized to default values at the time of creation of object, and remain accessible as long as the object is accessible.

**2. Local :** - These are the variables that are defined within a method. They remain accessible only

during the course of method execution. When the method finishes execution, these variables fall out of scope.

**3. Static:** - These are the class level variables. They are initialized when the class is loaded in JVM for the first time and remain there as long as the class remains loaded. They are not tied to any particular object instance.

**7. What happens if you don't initialize an instance variable of any of the primitive types in Java?**  
Java by default initializes it to the default value for that primitive type. For the integral types the default value is 0, for floating point data types default value is 0.0 and for the boolean types it is false.

## ***DURGA'S INTERVIEW QUESTIONS***

**8. What will be the initial value of an object reference, which is defined as an instance variable?**

The object references are all initialized to null in Java. However in order to do anything useful with these references, you must set them to a valid object, else you will get NullPointerExceptions everywhere you try to use such default initialized references.

**9. Are main, next, delete and exit keywords in Java?**

No, they are not keywords in Java. delete is not a keyword in Java. Java does not make use of explicit destructors the way C++ does. To exit a program explicitly you use exit method in System object.

**10. Is String a primitive data type in Java?**

No String is not a primitive data type in Java, even though it is one of the most extensively used object. Strings in Java are instances of String class defined in java.lang package.

**11. What does it mean that a method or field is "static"?**

static variables are class level variables. Only one copy will be created per class and shared by all the objects. We can access static variables either by class name or by object reference but recommended way is using class name. If you change the value of a static variable by using any object reference, it will automatically reflect for all the object references. That's how library methods like System.out.println() work out is a static field in the java.lang.System class. Static methods are class level methods. Mostly these are utility methods we can access without creating objects. Static methods can be referenced with the name of the class rather than the name of a particular object of the class (though that works too).

**12. Give a few reasons for using Java?**

Built-in support for multi-threading, socket communication, and memory management (automatic garbage collection). Object Oriented (OO). Better portability than other languages across operating systems. Supports Web based applications (Applet, Servlet, and JSP), distributed applications (sockets, RMI, EJB etc) and network protocols (HTTP, JRMP etc) with the help of extensive standardized APIs (Application Programming Interfaces).

**13. What are "static initializers" or "static blocks with no function names"?**

When a class is loaded, all blocks that are declared static and don't have function name (i.e. static initializers) are executed even before the constructors are executed. As the name suggests they are typically used to initialize static fields.

**14. How do you express an 'is a' relationship and a 'has a' relationship or explain inheritance and composition? What is the difference between composition and aggregation?**

The 'is a' relationship is expressed with inheritance and 'has a' relationship is expressed with composition. Both inheritance and composition allow you to place sub-objects inside your new

## ***DURGA'S INTERVIEW QUESTIONS***

class. Two of the main techniques for code reuse are class inheritance and object composition. Inheritance is uni-directional. For example House is a Building. But Building is not a House. Inheritance uses extends key word. Composition: is used when House has a Bathroom. It is incorrect to say House is a Bathroom. Composition simply means using instance variables that refer to other objects. The class House will have an instance variable, which refers to a Bathroom object

**15. What is the main difference between a String and a StringBuffer class?**

**String** is immutable: you can't modify a string object but can replace it by creating a new instance. Creating a new instance is rather expensive.

**StringBuffer** is mutable: use StringBuffer or StringBuilder when you want to modify the contents. StringBuilder was added in Java 5 and it is identical in all respects to StringBuffer except that it is not synchronized, which makes it slightly faster at the cost of not being thread-safe.

**16. What is the difference between processes and threads?**

A process is an execution of a program but a thread is a single execution sequence within the process. A process can contain multiple threads. A thread is sometimes called a lightweight process.

**17. Explain different ways of creating a thread?**

Threads can be used by either :

Extending the Thread class

Implementing the Runnable interface.

**18. What are transient variables in java?**

Transient variables are variable that cannot be serialized.

**19. What is synchronization?**

Synchronization is the ability to control the access multiple threads to shared resources.

Synchronization stops multithreading. With synchronization , at a time one thread will be able to access a shared resource.

**20. What is the difference between yield() and sleep()?**

When a object invokes yield() it returns to ready state for giving the chance to remaining threads of same priority. But when an object invokes sleep() method enters into sleeping state. After sleeping time expires then only thread entered into ready state

**21. What is finalize() method?**

Garbage Collector calls this method just before destroying any object to perform clean up activities.

## ***DURGA'S INTERVIEW QUESTIONS***

### **22.Can we call finalize() method ?**

Yes. Nobody will stop us to call any method , if it is accessible in our class. But a garbage collector cannot call an object's finalize method if that object is reachable.

### **23.Can we declare an anonymous class as both extending a class and implementing an interface?**

No. An anonymous class can extend a class or implement an interface but not both simultaneously.

### **24.What is the use of “bin” and “lib” in JDK?**

“bin” contains all tools such as javac, appletviewer, awt tool, etc. whereas “lib” contains API and all packages

### **25.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.

### **26.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.

### **27. What is final, finalize() and finally?**

**final** : final keyword can be used for class, method and variables. A final class cannot be subclassed and it prevents other programmers from subclassing a secure class to invoke insecure methods. A final method can't be overridden. A final variable can't change from its initialized value.

**finalize()** : finalize() method is used just before an object is destroyed and can be called just prior to garbage collection.

**finally** : finally, a key word used in exception handling, creates a block of code that will be executed after a try/catch block has completed and before the code following the try/catch block. The finally block will execute whether or not an exception is thrown. For example, if a method opens a file upon exit, then you will not want the code that closes the file to be bypassed by the exception-handling mechanism. This finally keyword is designed to address this contingency.

### **28.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. These exceptions must be handled by the caller.

## ***DURGA'S INTERVIEW QUESTIONS***

### **29. How many ways can an argument be passed to a subroutine?**

An argument can be passed in two ways. They are Pass by Value and Passing by Reference.

**Passing by value:** This method copies the value of an argument into the formal parameter of the subroutine.

**Passing by reference:** In this method, a reference to an argument (not the value of the argument) is passed to the parameter.

**30.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.

**31.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.

**32.How many methods in the Externalizable interface?**

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

**33. 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.

**34. 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.

**35. How do I serialize an object to a file?**

The class whose instances are to be serialized should implement an interface Serializable. Then you pass the instance to the ObjectOutputStream which is connected to a FileOutputStream. This will save the object to a file.

**36. Which methods of Serializable interface should I implement?**

The serializable interface is an empty interface, it does not contain any methods. So we do not implement any methods

## ***DURGA'S INTERVIEW QUESTIONS***

**37. How can I customize the seralization process? i.e. how can one have a control over the serialization process?**

Yes it is possible to have control over serialization process. The class should implement

Externalizable interface. This interface contains two methods namely read External and writeExternal. You should implement these methods and write the logic for customizing the serialization process.

**38. What is the common usage of serialization?**

Whenever an object is to be sent over the network, objects need to be serialized. Moreover if the state of an object is to be saved, objects need to be serialized.

**39. When you serialize an object, what happens to the object references included in the object?**

The serialization mechanism generates an object graph for serialization. Thus it determines whether the included object references are serializable or not. This is a recursive process. Thus when an object is serialized, all the included objects are also serialized alongwith the original object.

**40. What one should take care of while serializing the object?**

One should make sure that all the included objects are also serializable. If any of the objects is not serializable then it throws a NotSerializableException.

**41. What happens to the static fields of a class during serialization?**

Serialization ignores static fields, because they are not part of any particular state.

**42. Does importing a package imports the subpackages as well?**

No you will have to import the subpackages explicitly.

For eg: Importing com.MyTest.\* will import classes in the package MyTest only. It will not import any class in any of its subpackage.

**43. If I write return at the end of the try block, will the finally block still execute?**

Yes even if you write return as the last statement in the try block and no exception occurs, the finally block will execute. The finally block will execute and then the control return.

**44. Is it necessary that each try block must be followed by a catch block?**

It is not necessary that each try block must be followed by a catch block. It should be followed by either a catch block OR a finally block. And whatever exceptions are likely to be thrown should be declared in the throws clause of the method.

**45. What are the different ways to handle exceptions?**

There are two ways to handle exceptions,

1. **By using try-catch :** By wrapping the desired code in a try block followed by a catch block to

## ***DURGA'S INTERVIEW QUESTIONS***

catch the exceptions. and

2. **By using throws clause:** List the desired exceptions in the throws clause of the method and let the caller of the method handle those exceptions

**46.What are runtime exceptions?**

Runtime exceptions are those exceptions that are thrown at runtime because of either wrong input data or because of wrong business logic etc. These are not checked by the compiler at compile time.

**47.What are checked exceptions?**

Checked exception are those which the Java compiler forces you to catch for smooth execution of program at runtime. e.g. IOException are checked Exceptions.

**48.Does Java provide any construct to find out the size of an object?**

No there is not sizeof operator in Java. So there is not direct way to determine the size of an object directly in Java.

**49.What is Externalizable interface?**

Externalizable is an interface which contains two methods readExternal and writeExternal. These methods give you a control over the serialization mechanism. Thus if your class implements this interface, you can customize the serialization process by implementing these methods

**50.What is the purpose of garbage collection in Java, and when is it used?**

The purpose of garbage collection is to identify and discard objects that are no longer needed by a program so that their resources can be reclaimed and reused. A Java object is subject to garbage collection when it becomes unreachable to the program in which it is used.

**51.What is the difference between a constructor and a method?**

A constructor is a member function of a class that is used to create objects of that class. It has the same name as the class itself, has no return type, and is invoked using the new operator.

A method is an ordinary member function of a class. It has its own name, a return type (which may be void), and is invoked using the dot operator.

**52.What if the main method is declared as private?**

The program compiles properly but at runtime it will give “Main method not public.” message.

**53.What is an Iterator interface?**

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

**54.Describe the Garbage Collection process in Java ?**

The JVM spec mandates automatic garbage collection outside of the programmers control. The

## ***DURGA'S INTERVIEW QUESTIONS***

System.gc() or Runtime.getRuntime().gc() is merely a suggestion to the JVM to run the GC process but is NOT guaranteed.

**55.What is constructor chaining and how is it achieved in Java ?**

A child object constructor always first needs to construct its parent (which in turn calls its parent constructor.). In Java it is done via an implicit call to the no-args constructor as the first statement



**56.What methods can be overridden in Java?**

All Java methods can be overwritten in subclasses except those that are declared final, static, and private.

**57.What is daemon thread and which method is used to create the daemon thread?**

Daemon thread is a low priority thread which runs intermittently in the back ground for providing the support to Non-Daemon . setDaemon method is used to create a daemon thread.

**58.What are Transient and Volatile Modifiers?**

**Transient:** The transient modifier applies to variables only and it is not stored as part of its object's Persistent state. Transient variables are not serialized.

**Volatile:** Volatile modifier applies to variables only and it tells the compiler that the variable modified by volatile can be changed unexpectedly by other parts of the program. JVM maintains a separate copy of the volatile variable for every thread.

**59.What is the difference between instanceof and isInstance?**

instanceof is used to check to see if an object can be cast into a specified type without throwing a classCastException. isInstance() determines if the specified Object is assignment-compatible with the object represented by this Class. This method is the dynamic equivalent of the Java language instanceof operator. The method returns true if the specified Object argument is non-null and can be cast to the reference type represented by this Class object without raising a ClassCastException. It returns false otherwise.

**60.What does the “abstract” keyword mean in front of a method? A class?**

Abstract keyword declares either a method or a class. If a method has a abstract keyword in front of it, it is called abstract method. Abstract method has no body. It has only arguments and return type. Abstract methods act as placeholder methods that are implemented in the subclasses. Abstract classes can't be instantiated. If a class is declared as abstract, no objects of that class can be created. If a class contains any abstract method it must be declared as abstract

**61.What is JDBC? Describe the steps needed to execute a SQL query using JDBC.**

The JDBC is a pure Java API used to execute SQL statements. It provides a set of classes and interfaces that can be used by developers to write database applications.

The steps needed to execute a SQL query using JDBC:

## ***DURGA'S INTERVIEW QUESTIONS***

1. Open a connection to the database.
2. Execute a SQL statement.
3. Process the results.
4. Close the connection to the database.

**62.What are native methods? How do you use them?**

Native methods are methods whose implementation is provided in another programming language such as C. The main objective of native methods are to improve the performance of the system.

**63.How can you force all derived classes to implement a method present in the base class?**

Creating and implementing an interface would be the best way for this situation. Just create an interface with empty methods which forces a programmer to implement all the methods present under it. Another way of achieving this task is to declare a class as abstract with all its methods abstract.

**64.What's the difference between == and equals method?**

The equals method can be considered to perform a deep comparison of the value of an object, whereas the == operator performs a shallow comparison.

The equals() method compares the characters inside a string object. == operator compares two object references to check whether they refer to the same instances or not.

**65. How can you achieve Multiple Inheritance in Java?**

Java's interface mechanism can be used to implement multiple inheritance, with one important difference from c++ way of doing MI: the inherited interfaces must be abstract. This obviates the need to choose between different implementations, as with interfaces there are no implementations.

**66.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.

**67.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.

**68.What restrictions are placed on the values of each case of a switch statement?**

the switch argument is any type of variable is allowed which can be implicitly promoted to int type. The case labels must be compile time constants.

## ***DURGA'S INTERVIEW QUESTIONS***

**69.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. Associativity determines whether an expression is evaluated left-to-right or right-to-left

**70.What is a StringTokenizer ?**

StringTokenizer provide parsing process in which it identifies the delimiters provided by the user , by default delimiters are spaces, tab, newline etc. and separates them from the tokens. Tokens are those which are separated by delimiters.

**71.What are nested classes ?**

There are two types : static and non-static.

static class means the members in its enclosing class (class within class) can be accessed by creating an object and cannot be accessed directly without creating the object.

non-static class means inner class and can be accessed directly with the object created for the outer class no need to create again an object like static class.

**72.What is the difference between abstract class and interface?**

- a) All the methods declared inside an interface are abstract whereas abstract class may or may not contain abstract methods.
- b) In abstract class, key word abstract must be used for the methods whereas interface we need not use that keyword for the methods.
- c)Abstract class must have subclasses whereas interface can't have subclasses

**73.Under what conditions is an object's finalize() method invoked by the garbage collector?**

The garbage collector invokes an object's finalize() method just before destroying any method.

**74.What are the steps involved for making a connection with a database or how do you connect to a database?**

- a) Loading the driver :** To load the driver, Class. forName() method is used. Class. forName("sun.jdbc.odbc.JdbcOdbcDriver"); When the driver is loaded, it registers itself with the java. sql. DriverManager class as an available database driver.
- b) Making a connection with database:** To open a connection to a given database, DriverManager. getConnection() method is used. Connection con = DriverManager. getConnection ("jdbc:odbc:somedb", "user", "password");
- c) Executing SQL statements :** To execute a SQL query, java. sql. statements class is used. createStatement() method of Connection to obtain a new Statement object. Statement stmt = con. createStatement(); A query that returns data can be executed using the executeQuery() method of Statement. This method executes the statement and returns a java. sql. ResultSet that encapsulates the retrieved data: ResultSet rs = stmt. executeQuery("SELECT \* FROM some table");
- d) Process the results :** ResultSet returns one row at a time. Next() method of ResultSet object can be called to move to the next row. The getString() and getObject() methods are used for retrieving column values: while(rs. next()) { String event = rs. getString("event"); Object count = (Integer) rs. getObject("count").

## ***DURGA's INTERVIEW QUESTIONS***

**75.What is serialization and deserialization?**

Serialization is the process of writing the state of an object to a byte stream. Deserialization is the process of restoring these objects.

**76.What is the difference between JDBC and ODBC?**

- a) ODBC is for Microsoft and JDBC is for Java applications.
- b) ODBC can't be directly used with Java because it uses a C interface.
- c) ODBC makes use of pointers which have been removed totally from Java.
- d) ODBC mixes simple and advanced features together and has complex options for simple queries. But JDBC is designed to keep things simple while allowing advanced capabilities when required.
- e) ODBC requires manual installation of the ODBC driver manager and driver on all client machines. JDBC drivers are written in Java and JDBC code is automatically installable, secure, and portable on all platforms.
- f) JDBC API is a natural Java interface and is built on ODBC. JDBC retains some of the basic features of ODBC.

### 77.What is the difference between Reader/Writer and InputStream/Output Stream?

The Reader/Writer class is character-oriented and the InputStream/OutputStream class is byte-oriented.

### 78.What is a stream and what are the types of Streams and classes of the Streams?

A Stream is an abstraction that either produces or consumes information. There are two types of Streams and they are:

**Byte Streams:** Provide a convenient means for handling input and output of bytes.

**Character Streams:** Provide a convenient means for handling input & output of characters.

Byte Streams classes: Are defined by using two abstract classes, namely *InputStream* and *OutputStream*.

Character Streams classes: Are defined by using two abstract classes, namely *Reader* and *Writer*.

### 79.What are Vector, Hashtable, LinkedList and Enumeration?

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

**Hashtable :** The Hashtable class implements a Hashtable data structure. A Hashtable indexes and stores objects in a dictionary using hash codes as the object's keys. Hash codes are integer values that identify objects.

**LinkedList:** Removing or inserting elements in the middle of an array can be done using LinkedList. A LinkedList stores each object in a separate link whereas an array stores object references in consecutive locations.

**Enumeration:** An object that implements the Enumeration interface generates a series of elements, one at a time. It has two methods, namely *hasMoreElements()* and *nextElement()*.

*HasMoreElemnts()* tests if this enumeration has more elements and *nextElement* method returns successive elements of the series

## DURGA's INTERVIEW QUESTIONS

### 80.What is source and listener?

**Source :** A source is an object that generates an event. This occurs when the internal state of that object changes in some way.

**Listener :** A listener is an object that is notified when an event occurs. It has two major requirements. First, it must have been registered with one or more sources to receive notifications about specific types of events. Second, it must implement methods to receive and process these notifications.

### 81.What is the lifecycle of an applet?

**init()** method - Can be called when an applet is first loaded

**start()** method - Can be called each time an applet is started.

**paint()** method - Can be called when the applet is minimized or maximized.

**stop()** method - Can be used when the browser moves off the applet's page.

**destroy()** method - Can be called when the browser is finished with the applet.

**82.Can a for statement loop indefinitely?**

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

**83.What is the immediate superclass of the Applet class?**

Panel

**84.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 object is still reachable. However, an object's finalize() method may be invoked by other objects.

**85.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.

**86.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.

**87.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

**88.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.

## ***DURGA'S INTERVIEW QUESTIONS***

**89.What is connection pooling?**

With servlets, opening a database connection is a major bottleneck because we are creating and tearing down a new connection for every page request and the time taken to create connection will be more. Creating a connection pool is an ideal approach for a complicated servlet. With a connection pool, we can duplicate only the resources we need to duplicate rather than the entire servlet. A connection pool can also intelligently manage the size of the pool and make sure each connection remains valid. A number of connection pool packages are currently available. Some like DbConnectionBroker are freely available from Java Exchange Works by creating an object that dispenses connections and connection Ids on request. The ConnectionPool class maintains a Hashtable, using Connection objects as keys and Boolean values as stored values. The Boolean value indicates whether a connection is in use or not. A program calls getConnection() method of the

ConnectionPool for getting Connection object it can use; it calls returnConnection() to give the connection back to the pool.

#### 90.What are the types of statements in JDBC?

**Statement:** to be used createStatement() method for executing single SQL statement

**PreparedStatement** — To be used preparedStatement() method for executing same SQL statement over and over.

**CallableStatement** — To be used prepareCall() method for multiple SQL statements over and over.

#### 91.What is daemon thread and which method is used to create the daemon thread?

Daemon thread is a low priority thread which runs intermittently in the back ground doing the garbage collection operation for the java runtime system. setDaemon method is used to create a daemon thread.

#### 92. What is the class and interface in java to create thread and which is the most advantageous method?

Thread class and Runnable interface can be used to create threads and using Runnable interface is the most advantageous method to create threads because we need not extend thread class here.

#### 93. What are the methods for inter-thread communication and what is the class in which these methods are defined?

wait (), notify () and notifyAll() methods can be used for inter-thread communication and these methods are in Object class.

**wait()** : When a thread executes a call to wait() method, it surrenders the object lock and enters into a waiting state.

**notify() or notifyAll()** : To remove a thread from the waiting state, some other thread must make a call to notify() or notifyAll() method on the same object.

#### 94.What is multithreading?

Multithreading is the mechanism in which more than one thread run independent of each other within the process.

#### 95. What is the difference between process and thread?

Process is a program in execution whereas thread is a separate path of execution in a program.  
What is the difference between Array and vector?

Array is a set of related data type and static whereas vector is a growable array of objects and dynamic.

**96. What is the difference between String and StringBuffer?**

String objects are constants and immutable whereas StringBuffer objects are not.

b) String class supports constant strings whereas StringBuffer class supports growable and modifiable strings.

**97. What are inner class and anonymous class?**

**Inner class** : classes defined in other classes, including those defined in methods are called inner classes. An inner class can have any accessibility including private. Anonymous class : **Anonymous class** is a class defined inside a method without a name and is instantiated and declared in the same place and cannot have explicit constructors.

**98 . Can you have an inner class inside a method and what variables can you access?**

Yes, we can have an inner class inside a method and final variables can be accessed.

**99. What is a cloneable interface and how many methods does it contain?**

It is not having any method because it is a TAGGED or MARKER interface.

**100. How many times may an object's finalize() method be invoked by the garbage collector?**

An object's finalize() method may only be invoked once by the garbage collector.

**101. What modifiers may be used with top-level class?**

public, abstract and final,strictfp can be used for top-level class.

**102. 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.