

Soal Code B

Q: 61 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.

Q: 62 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);

Q: 63 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

Q: 64 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.

Q: 65 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.

Q: 66 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

Q: 67 Which three code fragments, added individually at line 29, produce the output 100? (Choose three.)

Soal Code B

```

10. class Inner {
11.     private int x;
12.     public void setX( int x ) { this.x = x;
13. }
14.     public int getX() { return x; }
15. }
16. class Outer {
17.     private Inner y;
18.     public void setY( Inner y ) { this.y =
19. y; }
20.     public Inner getY() { return y; }
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);

Q: 68 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

Q: 69 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.

Q: 70

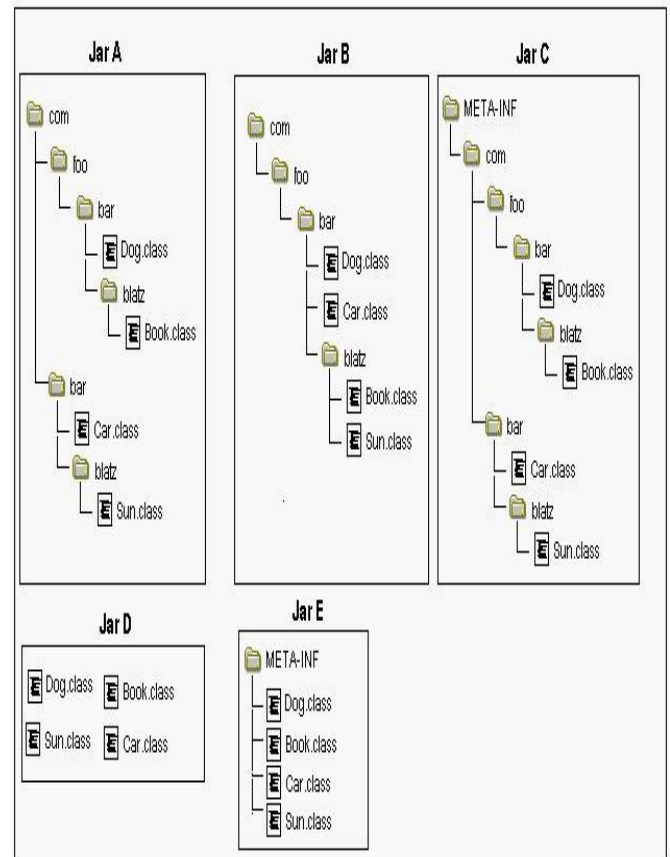
Given the fully-qualified class names:

```

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

Q: 71 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:

Soal Code B

```
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

Q: 72 Given a class Repetition:

```
1. package utils;
2.
3. public class Repetition {
4.     public static String twice(String s) {
5.         return s + s; }
6. }
```

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;

Q: 73 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.

Q: 74 Given:

```
11. static class A {
```

```
12.     void process() throws Exception { throw
13.         new Exception(); }
14. }
15. static class B extends A {
16.     void process() { System.out.println("B
17. "); }
18. }
19. public static void main(String[] args) {
20.     A a = new B();
21.     a.process();
22. }
```

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.

Q: 75 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.

Q: 76 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.

Q: 77 Given:

```
11. public static Iterator reverse(List list)
12.     {
13.         Collections.reverse(list);
14.         return list.iterator();
15.     }
16. public static void main(String[] args) {
```

Soal Code B

```
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.

Q: 78 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.

Q: 79 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.

Q: 80 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.

Q: 81 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

Q: 82 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`

Soal Code B

Q: 83 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

Q: 84 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.

Q: 85 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.

Q: 86 Given:

```
1. public class TestString3 {
2. public static void main(String[] args) {
3. // insert code here
4. System.out.println(s);
5. }
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).inse`
`rt(1, "24");`

Q: 87

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

3	3.141593	true	Math.PI
2	2.718282	false	Math.E

Q: 88 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

Q: 89 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?

Soal Code B

- 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.

Q: 90 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.

Q: 91

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) {
    
    for ( String dir : directories ) {
        
    }
    
    
}
```

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>

Q: 92 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();`

Q: 93 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.

Q: 94 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));`

Q: 95 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];
```

Soal Code B

```
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.

Q: 96 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();

Q: 97 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);

Q: 98 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.

Q: 99 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");

Q: 100 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.

Q: 101 What is the outcome of the code?

Soal Code B

```
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

Q: 102 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.

Q: 103 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.         Foo foo = new Foo( 300 );
26.         System.out.print( foo.getX() + "-" );
27.
28.         Foo fooFoo = fooBar( foo );
29.         System.out.print( foo.getX() + "-" );
30.         System.out.print( fooFoo.getX() + "-"
);
31.
32.         foo = fooBar( fooFoo );
33.         System.out.print( foo.getX() + "-" );
34.         System.out.print( fooFoo.getX() );
35.     }
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

Q: 104 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

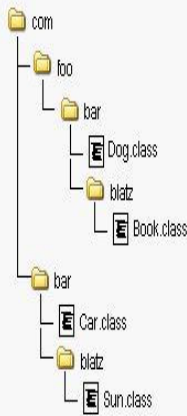
Q: 105

Soal Code B

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.

All 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.*;	package com.foo.bar.blatz;
import com.bar.*;	import com.*;
package com.bar;	package com;
import com.foo.*;	// blank
import com.foo.bar.*;	import com.foo.bar.Book;

Done

Q: 106

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.

Q: 107 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.

Q: 108 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

Q: 109 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.

Q: 110 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);

```

Soal Code B

```
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.

Q: 111 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)

Q: 112 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();
14.           iter.hasNext(); ) {
15.         int i = ((Integer)iter.next()).intValue();
16.         sum += i;
17.     }
18.     return sum;
19. }
```

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)"

Q: 113

```
Given: NumberNames nn = new NumberNames();
nn.put("one", 1);
System.out.println(nn.getNames());
```

Place the code into position to create a class that maps from Strings to integer values. The result of execution must be [one]. Some options may be used more than once.

```
public class NumberNames {
    private HashMap< Place here , Place here > map =
        new HashMap< Place here , Place here Place here ;

    public void put(String name, int value) {
        map.put( Place here , Place here );
    }
    public Place here getNames() {
        return map.keySet();
    }
}
```

Code

Set<int>	Set<Integer>	HashSet		
Set<Integer,String>	Set<int, String>	Set<String,Integer>		
Set<String,int>	Set<String>	NumberNames		
String	Integer	int	>	Done
>()	name	value	map	

Q: 114 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() <
17.         min.doubleValue()) min = added;
18.     if (max == null || added.doubleValue() >
19.         max.doubleValue()) 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> {

Q: 115 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("aAa");
7.         strings.add("AaA");
8.         strings.add("aAa");
9.         strings.add("AAaa");
10. Collections.sort(strings);
11.     for (String s : strings) {
12.         System.out.print(s + " ");
13.     }
```

Soal Code B

```
13. }
```

What is the result?

- A. Compilation fails.
- B. aAa aAa AAaa AaA
- C. AAaa AaA aAa aAa
- D. AaA AAaa aAaA aAa
- E. aAa AaA aAaA AAaa
- F. An exception is thrown at runtime.

Q: 116 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 class Circle implements 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 */ }
```

Q: 117 Given

```
10. class Foo {
11.     static void alpha() { /* more code here
*/ }
12.     void beta() { /* more code here */ }
13. }
```

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().

Q: 118 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.

Q: 119 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.

Q: 120 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