

Soal Code D

Q: 181 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.

Q: 182 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.

Q: 183

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

Q: 184 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);`

Q: 185 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.

Q: 186 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){/*more 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*/}`

Soal Code D

Q: 187

Place the code in the appropriate places such that this program will always output [1, 2].

```
import java.util.*;

public class MyInt {
    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); }
}

int i {
    MyInt i2 = (MyInt)o;
    return ;
}
```

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

Q: 188

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

Q: 189 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.

Q: 190 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.

Q: 191

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
Pet extends T
?
T
<?>
Pet

Done

Q: 192 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.

Soal Code D

Q: 193 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> {

Q: 194 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>());

Q: 195

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

Q: 196 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

Q: 197 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.

Q: 198 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).

Soal Code D

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

Q: 199 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: 200

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;`

Q: 201 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

Q: 202 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: 203 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

Soal Code D

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

Q: 204 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.

Q: 205 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: 206 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)

Q: 207

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?

```
1. public class A {
2.
3.     private int counter = 0;
4.
5.     public static int getInstanceCount() {
6.         return counter;
7.     }
8.
9.     public A() {
10.        counter++;
11.    }
12.
13. }
```

- 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.

Q: 208 What is the result?

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


Soal Code D

- A. go in Goban
go in Sente
- B. go in Sente
go in Goban
- C. go in Goban
go in Sente
- D. Compilation fails because of an error in line 17.

Q: 209 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.

Q: 210 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.

Q: 211 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.

Q: 212 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

Q: 213 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.

Q: 214 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. }
```

Soal Code D

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.

Q: 215 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

Q: 216 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.

Q: 217

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 {
```

	Methods
Place here	private void bar(int x) {}
Place here	public void bar(int x) {}
Place here	public int bar(String x) { return 1; }
Place here	public Alpha bar(int x) {}
Place here	public void bar(int x, int y) {}
Place here	public int bar(int x) { return x; }

Q: 218 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)`

Q: 219 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();

Q: 220

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

Q: 221 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. }
```

Soal Code D

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.

Q: 222

```
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
Place here	(no more output)

Done

Q: 223

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();
        Place here
        Place here
    }
    public void run() {
        System.out.print("Run. ");
    }
    public void doIt() {
        System.out.print("doIt. ");
    }
}
```

Code Elements

t.start();	t.join();	t.pause(10);	run();
t.run();	t.doIt();	doIt();	

Done

Q: 224 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.

Q: 225 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.             } }
```

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 }

Q: 226 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. }
```


Soal Code D

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.

Q: 227 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: 228

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("Fizylylifting");
15.        modifyDesc(it,
"Scrumdiddlyumptious");
16.
System.out.println(it.getDescription());
17.
System.out.println(it2.getDescription());
18.    }
19. }
```

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

Q: 229 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: 230 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: 231 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

Soal Code D

Q: 232 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: 233 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

Q: 234 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;

Q: 235 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].

Q: 236 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)

Soal Code D

Q: 237 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.

Q: 238

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

Q: 239 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: 240 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