

Oracle

Exam 1z0-851

Java Standard Edition 6 Programmer Certified Professional Exam

Version: Demo

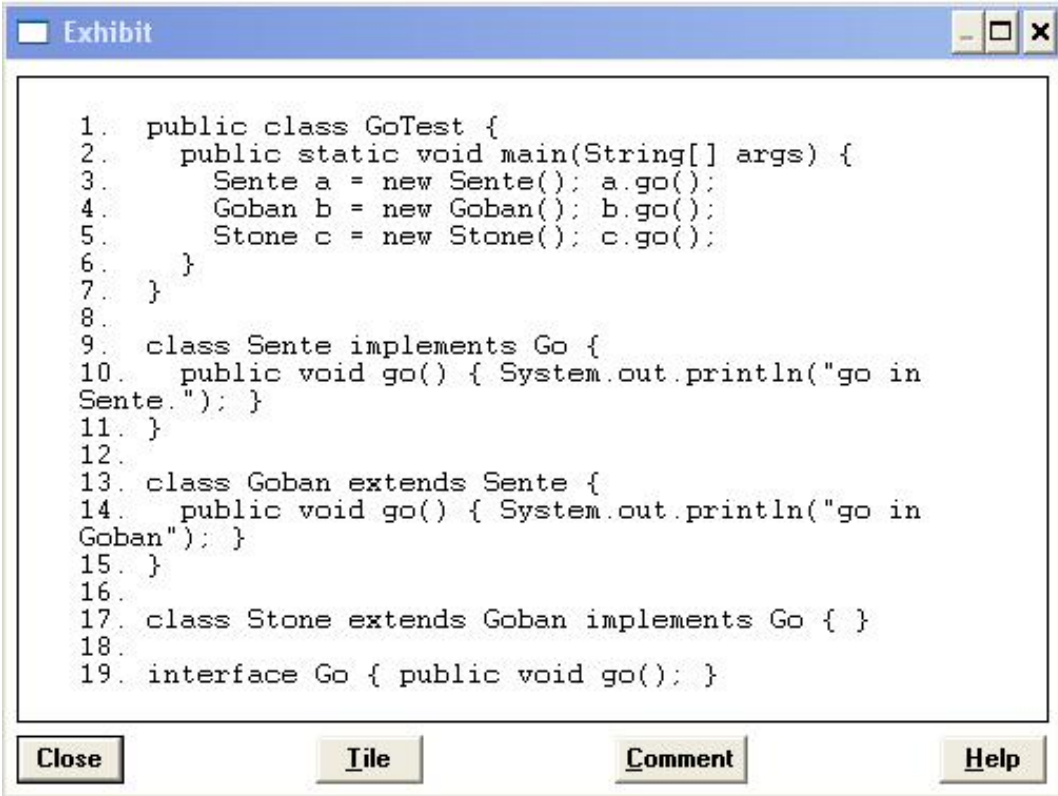
[Total Questions: 10]

Topic break down

Topic	No. of Questions
Topic 1: Volume A	5
Topic 2: Volume B	5

Topic 1, Volume A**Question No : 1 - (Topic 1)**

Click the Exhibit button. What is the result?



```
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(); }
```

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

Answer: C

Question No : 2 - (Topic 1)

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 No : 3 - (Topic 1)

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

Question No : 4 - (Topic 1)

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 No : 5 - (Topic 1)

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

Topic 2, Volume B

Question No : 6 - (Topic 2)

Given:

```
1. class TestException extends Exception { }  
2. class A {  
3.     public String sayHello(String name) throws TestException {  
4.         if(name == null) throw new TestException();  
5.         return "Hello " + name;  
6.     }
```

```
7. }  
8. public class TestA {  
9.     public static void main(String[] args) {  
10.         new A().sayHello("Aiko");  
11.     }  
12. }
```

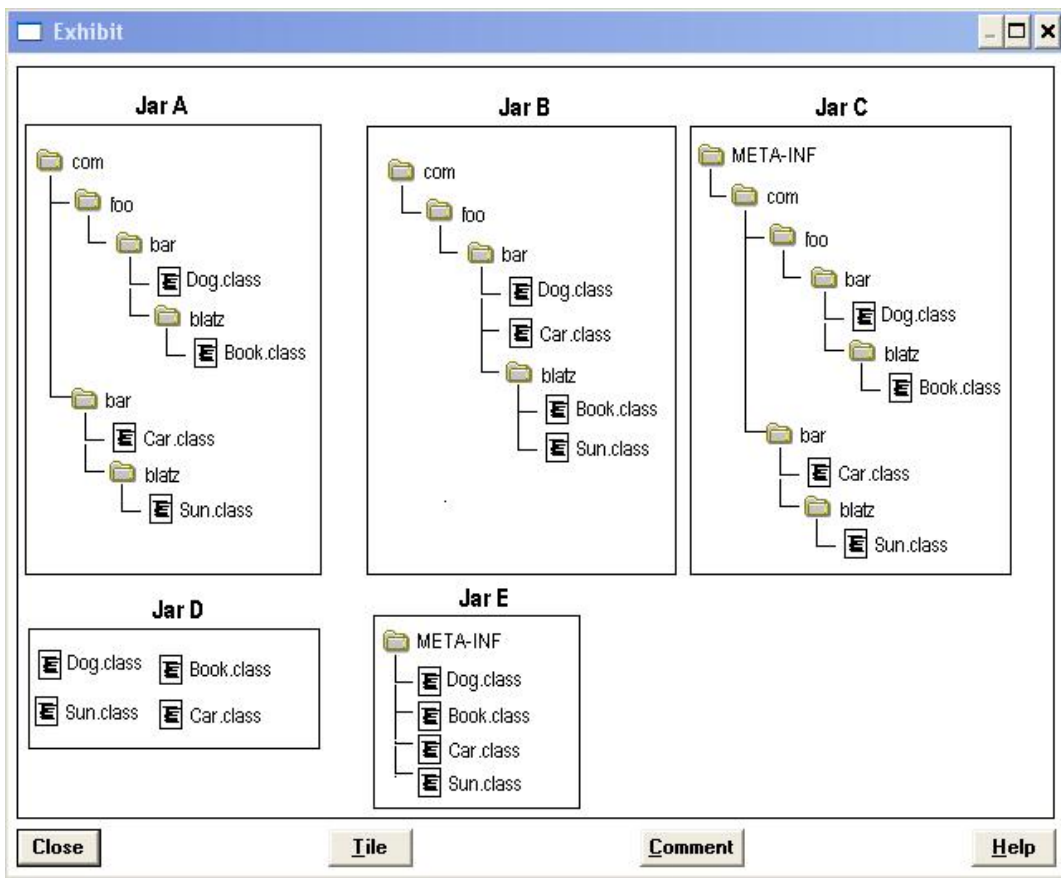
Which statement is true?

- A. Compilation succeeds.
- B. Class A does not compile.
- C. The method declared on line 9 cannot be modified to throw `TestException`.
- D. `TestA` compiles if line 10 is enclosed in a `try/catch` block that catches `TestException`.

Answer: D

Question No : 7 - (Topic 2)

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

Question No : 8 - (Topic 2)

Given:

3. class Employee {
4. String name; double baseSalary;
5. Employee(String name, double baseSalary) {
6. this.name = name;
7. this.baseSalary = baseSalary;

8. }

9. }

10. public class SalesPerson extends Employee {

11. double commission;

12. public SalesPerson(String name, double baseSalary, double commission) {

13. // insert code here

14. }

15. }

Which two code fragments, inserted independently at line 13, will compile? (Choose two.)

A. super(name, baseSalary);

B. this.commission = commission;

C. super();

this.commission = commission;

D. this.commission = commission;

super();

E. super(name, baseSalary);

this.commission = commission;

F. this.commission = commission;

super(name, baseSalary);

G. super(name, baseSalary, commission);

Answer: A,E

Question No : 9 DRAG DROP - (Topic 2)

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

Answer:

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

Modifiers

final
protected
private
abstract
static

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

Modifiers

final
protected
private
abstract
static

Question No : 10 - (Topic 2)

Given that the elements of a PriorityQueue are ordered according to natural ordering, and:

```
2. import java.util.*;
3. public class GetInLine {
4.     public static void main(String[] args) {
5.         PriorityQueue<String> pq = new PriorityQueue<String>();
6.         pq.add("banana");
7.         pq.add("pear");
8.         pq.add("apple");
9.         System.out.println(pq.poll() + " " + pq.peek());
10.    }
11. }
```

What is the result?

- A. apple pear
- B. banana pear
- C. apple apple
- D. apple banana
- E. banana banana

Answer: D