

Exam : SUN 212-055

**Title : Sun Certified Programmer
for the Java 2 Platform.SE
5.0**

Version : Demo

Important Note, Please Read Carefully

Other VisualExams products

[All visualexams IT Exam Products](#)

Our products of Offline Testing Engine

Use the offline Testing engine product to practice the questions in an exam environment.

Build a foundation of knowledge which will be useful also after passing the exam.

[visualexams Testing Engine](#)

Latest Version

We are constantly reviewing our products. New material is added and old material is revised. Free updates are available for 90 days after the purchase. You should check your member zone at VisualExams and update 3-4 days before the scheduled exam date.

Here is the procedure to get the latest version:

1. Go to <http://www.visualexams.com/>
2. Log in the **User Center**
3. The latest versions of all purchased products are downloadable from here. Just click the links.

Feedback

If you spot a possible improvement then please let us know. We always interested in improving product quality.

Feedback should be send to Visualexams@hotmail.com. You should include the following: Exam number, version, page number, question number, and your login Account.

Our experts will answer your mail promptly.

Explanations

This product does not include explanations at the moment. If you are interested in providing explanations for this exam, please contact Visualexams@hotmail.com.

Features

- Comprehensive questions with complete details
- Instant Downloadable in PDF form.
- Verified Answers Researched by Industry Experts
- Questions accompanied by exhibits.
- Drag and Drop questions as experienced in the Actual Exams.
- These questions and answers are backed by our GUARANTEE.
- Questions updated on regular basis.
- Like actual certification exams our product is in multiple-choice questions (MCQs)

Commitment to Your Success: At VisualTestExam.com, we are committed to your ongoing success. Our exams and questions are constantly being updated and compared to industry standards.

Want to earn a Microsoft certification like MCSE, MCSE 2003, CCNA, CCNP? Thinking about getting an A+ or CCSP?

A, CCSP or Network+ Certification with less effort and time. You will be astonished at the theoretical and practical knowledge you will acquire in such a short period of time using our Certification Training Products. Our Study material will enable you to pass your Microsoft, Your Cisco and any other certification on the very first attempt.

Guarantee

Visualexams provides the most competitive quality of all exams for the customers, we guarantee your success at the first attempt with only our Certification Question&Answers, if somehow you do not pass the exam at the first time, we will not only arrange FULL REFUND for you, but also provide you another exam of your claim, ABSOLUTELY FREE!

1. 現有:

```
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. }
```

結果為何?

- A. 0.0
- B. 編譯失敗。
- C. 在執行時期 parse 方法會丟出一個 ParseException。
- D. 在執行時期 parse 方法會丟出一個 NumberFormatException。

Answer: B

2. 現有:

```
10. int x = 0;  
11. int y = 10;  
12. do {  
13.     y--;  
14.     ++x;  
15. } while (x < 5);  
16. System.out.print(x + "," + y);
```

結果為何?

- A. 5,6

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

Answer: B

3. 現有:

- ```
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. }
```

結果為何?

- A. short LONG
- B. SHORT LONG
- C. 編譯失敗。
- D. 在執行時期丟出了一個例外。

**Answer: C**

4. 現有:

- ```
11. String test = "This is a test";  
12. String [] tokens = test.split("\s");  
13. System.out.println(tokens.length);
```

結果為何?

- A. 0

- B. 1
- C. 4
- D. 編譯失敗。
- E. 在執行時期丟出了一個例外。

Answer: D

5. 現有：

```
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; }  
}
```

應該做哪三項修改，才能調整這個類別，以供多個執行緒安全地使用？（選擇三項。）

- A. 用 `synchronized` 關鍵字宣告 `reset()`
- B. 用 `synchronized` 關鍵字宣告 `getName()`
- C. 用 `synchronized` 關鍵字宣告 `getCount()`
- D. 用 `synchronized` 關鍵字宣告建構元
- E. 用 `synchronized` 關鍵字宣告 `increment()`

Answer: ACE

6. 現有：

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

以及：

```
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();

哪三項是正確的？（選擇三項。）

A. p0 = p1;

B. p1 = p2;

C. p2 = p4;

D. p2 = (ClassC)p1;

E. p1 = (ClassB)p3;

F. p2 = (ClassC)p4;

Answer: AEF

7. 現有:

```
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: }
```

以及:

```
30: Hello c = new Hello(5);
```

```
31: System.out.println(c.title);
```

結果為何?

- A. Hello
- B. Hello World
- C. 編譯失敗。
- D. Hello World 5
- E. 這個程式碼可以執行，但沒有輸出。
- F. 在執行時期丟出了一個例外。

Answer: C

8. 現有：

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); }

結果為何？

- A. 檔案可以編譯，而沒有錯誤。
- B. 編譯失敗。只有第 7 行有一個錯誤。
- C. 編譯失敗。只有第 12 行有一個錯誤。
- D. 編譯失敗。只有第 13 行有一個錯誤。
- E. 編譯失敗。只有第 7 行與第 12 行有錯誤。
- F. 編譯失敗。只有第 7 行與第 13 行有錯誤。
- G. 編譯失敗。第 7、12、與 13 行有錯誤。

Answer: A

9. 現有:

10. package com.sun.scjp;

11. public class Geodetics {

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

13. }

哪兩項可以正確地存取 Geodetics 類別的 DIAMETER 成員? (選擇兩項。)

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: AC

10. 現有:

10. class Nav{

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

12. }

13. public class Sprite{

14. // insert code here

15. }

第 14 行加入哪一項程式碼後，可讓 `Sprite` 類別編譯？

- A. `Direction d = NORTH;`
- B. `Nav.Direction d = NORTH;`
- C. `Direction d = Direction.NORTH;`
- D. `Nav.Direction d = Nav.Direction.NORTH;`

Answer: D

11. 現有：

```
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. }
```

第 15 行插入哪一項程式碼後，可讓 `Sprite` 類別編譯？

- 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

12. 現有：

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

某程式設計師想要建立一個稱為 `B` 的介面，並以 `A` 為其父類別。

哪一項介面的宣告是正確的？

- 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

13. 現有：

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. }

結果為何？

- A. TestA
- B. TestB
- C. 編譯失敗。
- D. 在執行時期丟出了一個例外。

Answer: B

14. 現有：

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. }

結果為何？

A. test

B. null

C. 在執行時期丟出了一個例外。

D. 第 1 行的一個錯誤會造成編譯失敗。

E. 第 4 行的一個錯誤會造成編譯失敗。

F. 第 5 行的一個錯誤會造成編譯失敗。

Answer: A

15. 現有：

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. }

以及一個延展並完全實行 Shape 類別的 Circle 類別。

哪一項是正確的？

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

16. 現有：

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. }

在第 17 行可以各自分別插入哪兩種方法，以便正確地完成 **Sales** 類別？（選擇兩項。）

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: BD

17. 現有：

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

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

哪一個類別正確地使用 Data 介面與 Info 類別？

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

18. 現有：

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. }
```

哪兩個類別正確地使用 **Shape** 類別？（選擇兩項。）

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: BE

19. 哪兩個類別可以正確地實行 `java.lang.Runnable` 以及 `java.lang.Cloneable` 兩種介面？（選擇兩項。）

A. `public class Session`

```
implements Runnable, Cloneable {  
    public void run();  
    public Object clone();  
}
```

B. `public class Session`

```
extends Runnable, Cloneable {  
    public void run() { /* do something */ }  
    public Object clone() { /* make a copy */ }  
}
```

C. `public class Session`

```
implements Runnable, Cloneable {  
    public void run() { /* do something */ }  
    public Object clone() { /* make a copy */ }  
}
```

D. `public abstract class Session`

```
implements Runnable, Cloneable {  
    public void run() { /* do something */ }  
    public Object clone() { /*make a copy */ }  
}
```

E. `public class Session`

```
implements Runnable, implements Cloneable {  
    public void run() { /* do something */ }  
    public Object clone() { /* make a copy */ }  
}
```

Answer: CD

20. 現有：

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

哪三項在第 12 行有效？（選擇三項。）

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

Answer: ABD

Visualexams.com was founded in 2006. The safer,easier way to help you pass any IT Certification exams . We provide high quality IT Certification exams practice questions and answers(Q&A). Especially Adobe, Apple, Citrix, Comptia, EMC, HP, HuaWei, LPI, Nortel, Oracle, SUN, Vmware and so on. And help you pass any IT Certification exams at the first try.

Web site: <http://www.visualexams.com>

You can reach us at any of the email addresses listed below.

E-Mail: [visualexams \(at\) hotmail.Com](mailto:visualexams@hotmail.com)