

Exam : **310-065Big5**

Title : Sun Certified Programmer
for the Java 2 Platform, SE
6.0

Version : Demo

www.Pass4itSure.com

1. 已知:

35. String #name = "Jane Doe";

36. int \$age = 24;

37. Double _height = 123.5;

38. double ~temp = 37.5;

以下敘述哪兩個正確? (請選擇兩個答案。)

A. 第 35 行無法編譯。

B. 第 36 行無法編譯。

C. 第 37 行無法編譯。

D. 第 38 行無法編譯。

Answer: AD

2. 已知:

11. public static Iterator reverse(List list) {

12. Collections.reverse(list);

13. return list.iterator();

14. }

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

16. List list = new ArrayList();

17. list.add("1"); list.add("2"); list.add("3");

18. for (Object obj: reverse(list))

19. System.out.print(obj + ", ");

20. }

結果為何?

A. 3, 2, 1,

B. 1, 2, 3,

C. 編譯失敗。

D. 程式碼順利執行, 但沒有輸出。

E. 執行階段丟出異常。

Answer: C

3. 已知:

```
5. class A {  
6.     void foo() throws Exception { throw new Exception(); }  
7. }  
8. class SubB2 extends A {  
9.     void foo() { System.out.println("B "); }  
10. }  
11. class Tester {  
12.     public static void main(String[] args) {  
13.         A a = new SubB2();  
14.         a.foo();  
15.     }  
16. }
```

結果為何?

- A. B
- B. B, 後面是 Exception。
- C. 編譯會失敗, 因為第9 行有錯誤。
- D. 編譯會失敗, 因為第4 行有錯誤。
- E. 丟出 Exception, 沒有其他輸出。

Answer: D

4. 已知:

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

將以下哪兩個程式碼片段個別插入第 3 行後, 會輸出 4247? (請選擇兩個答案。)

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

Answer: BE

5. 已知:

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

如果建立 Beagle 類別的實例，然後建立 Serialized，接著建立 deSerialized，則會有什麼結果？

- A. ad B.  
ada C.  
add D.  
adad
- E. 編譯失敗。  
F. 執行階段丟出異常。

**Answer:** B

6. 已知:

```
1. public class LineUp {
2. public static void main(String[] args) {
3. double d = 12.345;
4. // insert code here
5. }
6. }
```

將以下哪個程式碼片段插入第 4 行後，會輸出 | 12.345|?

- A. System.out.printf("|%7d|\n", d);
- B. System.out.printf("|%7f|\n", d);
- C. System.out.printf("|%3.7d| \n", d);
- D. System.out.printf("|%3.7f| \n", d);
- E. System.out.printf("|%7.3d|\n", d);
- F. System.out.printf("|%7.3f|\n", d);

**Answer: F**

7. 以下哪三個程式碼可順利編譯並執行，而不會產生異常？（請選擇三個答案。）

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

**Answer: CEF**

8. 一組程式設計師負責審核新公用程式類別的 API 提案。他們討論過後，發現可以減少 API 中的方法數

量，卻不會減損功能。

如果他們實作新設計，必須要提倡哪兩項物件導向原則？

- A. 更寬鬆連結
- B. 更緊密連結
- C. 更低內聚力
- D. 更高內聚力
- E. 更脆弱封裝
- F. 更堅強封裝

**Answer: A**

9. 已知：

```
2. public class Hi {
3. void m1() {}
4. protected void() m2 {}
5. }
6. class Lois extends Hi {
7. // insert code here
8. }
```

將以下哪四個程式碼片段個別插入第 7 行後即可順利編譯？(請選擇四個答案。)

- A. public void m1() {}
- B. protected void m1() {}
- C. private void m1() {}
- D. void m2() {}
- E. public void m2() {}
- F. protected void m2() {}
- G. private void m2() {}

**Answer: AB EF**

10. 已知:

```
1. import java.util.*;
2. public class Example {
3. public static void main(String[] args) {
4. // insert code here
5. set.add(new Integer(2));
6. set.add(new Integer(1));
7. System.out.println(set);
8. }
9. }
```

請問要在第 4 行插入哪個程式碼，才能保證此程式會輸出 [1, 2] ?

- A. Set set = new TreeSet();
- B. Set set = new HashSet();
- C. Set set = new SortedSet();
- D. List set = new SortedList();
- E. Set set = new LinkedHashSet();

**Answer: A**

11. 已知:

```
1. import java.util.*;
2.
3. public class LetterASort{
4. public static void main(String[] args) {
5. ArrayList<String> strings = new ArrayList<String>();
6. strings.add("aAaA");
7. strings.add("AaA");
8. strings.add("aAa");
9. strings.add("AAaa");
10. Collections.sort(strings);
```

- ```
11. for (String s : strings) { System.out.print(s + " "); }  
12. }  
13. }
```

結果為何？

- A. 編譯失敗。
- B. aAaA aAa AAaa AaA
- C. AAaa AaA aAa aAaA
- D. AaA AAaa aAaA aAa
- E. aAa AaA aAaA AAaa
- F. 執行階段丟出異常。

Answer: C

12. 已知下列目錄結構：

bigProject

```
|--source  
|   |--Utils.java  
|  
|--classes  
    |--
```

以及下列指令行呼叫：

```
javac -d classes source/Utils.java
```

假設目前的目錄是 bigProject，則會有什麼結果？

- A. 如果編譯成功，則會把Utils.class 加入來源目錄。
- B. 編譯器會傳回無效的旗標錯誤。
- C. 如果編譯成功，則會把Utils.class 加入類別目錄。
- D. 如果編譯成功，則會把Utils.class 加入 bigProject 目錄。

Answer: C

13. 已知：

```
10. package com.sun.scjp;
```



```
11. public class Geodetics {  
12.     public static final double DIAMETER = 12756.32; // kilometers  
13. }
```

哪兩項能正確存取 Geodetics 類別的 DIAMETER 成員？(請選擇兩個答案。)

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

14. 已知:

```
10. class Nav{  
11.     public enum Direction { NORTH, SOUTH, EAST, WEST }  
12. }  
13. public class Sprite{  
14.     // insert code here  
15. }
```

請問要在第 14 行插入哪個程式碼，才能讓 Sprite 類別進行編譯？

A. Direction d = NORTH;

B. Nav.Direction d = NORTH;

C. Direction d = Direction.NORTH;

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

Answer: D

15. 已知:

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. }

請問要在第 15 行插入哪個程式碼，才能讓 Sprite 類別進行編譯?

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

16. 按一下 [範例] 按鈕。

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. }
```

以下敘述哪三個正確? (請選擇三個答案。)

- A. 編譯失敗。
- B. 可編譯此程式碼，輸出為2。
- C. 如果移除第 16、17 和 18 行，編譯就會失敗。
- D. 如果移除第 24、25 和 26 行，編譯就會失敗。
- E. 如果移除第 16、17 和 18 行，即可編譯此程式碼，輸出為 2。
- F. 如果移除第 24、25 和 26 行，即可編譯此程式碼，輸出為 1。

Answer: BEF

17. 已知:

```
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. 執行階段丟出異常。

Answer: B

18. 已知:

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. 執行階段丟出異常。
- D. 編譯會失敗，因為第 行有錯誤。
- E. 編譯會失敗，因為第 行有錯誤。
- F. 編譯會失敗，因為第 行有錯誤。

Answer: A

19. 已知:

```
11. public interface A111 {
```

```
12.     String s = "yo";
```

```
13.     public void method1();
```

```
14. }
```

```
17. interface B { }
```

```
20. interface C extends A111, B {
```

```
21.     public void method1();
```

```
22.     public void method1(int x);
```

```
23. }
```

結果為何?

- A. 編譯成功。
- B. 編譯會失敗, 因為有多個錯誤。
- C. 編譯會失敗, 只有第0 行有錯誤。
- D. 編譯會失敗, 只有第1 行有錯誤。
- E. 編譯會失敗, 只有第2 行有錯誤。
- F. 編譯會失敗, 只有第 行有錯誤。

Answer: A

20. 已知:

```
11. public interface A { public void m1(); }
```

```
12.
```

```
13. class B implements A { }
```

```
14. class C implements A { public void m1() { } }
```

```
15. class D implements A { public void m1(int x) { } }
```

```
16. abstract class E implements A { }
```

```
17. abstract class F implements A { public void m1() { } }
```

```
18. abstract class G implements A { public void m1(int x) { } }
```

結果為何?

- A. 編譯成功。

- B. 「無法剛好一個類別
進行編譯。 C. 「無法剛好兩個類別
進行編譯。 D. 「無法剛好四個類別
進行編譯。 E. 「無法剛好三個類別
進行編譯。

Answer: C

www.Pass4itSure.com

Trying our product !

- ★ **100%** Guaranteed Success
- ★ **100%** Money Back Guarantee
- ★ **365 Days** Free Update
- ★ **Instant Download** After Purchase
- ★ **24x7** Customer Support
- ★ Average **99.9%** Success Rate
- ★ More than **69,000** Satisfied Customers Worldwide
- ★ Multi-Platform capabilities - **Windows, Mac, Android, iPhone, iPod, iPad, Kindle**

Need Help

Please provide as much detail as possible so we can best assist you.

To update a previously submitted ticket:



Submit A Ticket

One Year Free Update



Free update is available within One Year after your purchase. After One Year, you will get 50% discounts for updating. And we are proud to boast a 24/7 efficient Customer Support system via Email.



Money Back Guarantee

To ensure that you are spending on quality products, we provide 100% money back guarantee for 30 days from the date of purchase.



Security & Privacy

We respect customer privacy. We use McAfee's security service to provide you with utmost security for your personal information & peace of mind.

Guarantee & Policy | Privacy & Policy | Terms & Conditions

Any charges made through this site will appear as Global Simulators Limited.

All trademarks are the property of their respective owners.

Copyright © 2004-2014, All Rights Reserved.