

T estpassport問題集



更に上のクオリティ 更に上のサービス

一年で無料進級することに提供する
[Http://www.testpassport.jp](http://www.testpassport.jp)

Exam : **1Z1-803**

Title : **Java SE 7 Programmer I**

Version : **DEMO**

1. Given the code fragment:

```
int [][] array2D = {{0, 1, 2}, {3, 4, 5, 6}};  
system.out.print (array2D[0].length+ "" );  
system.out.print(array2D[1].getClass().isArray() + "");  
system.out.println (array2D[0][1]);
```

What is the result?

- A. 3false1
- B. 2true3
- C. 2false3
- D. 3true1
- E. 3false3
- F. 2true1
- G. 2false1

Answer: D

Explanation: The length of the element with index 0, {0, 1, 2}, is 3. Output: 3

The element with index 1, {3, 4, 5, 6}, is of type array. Output: true

The element with index 0, {0, 1, 2} has the element with index 1: 1. Output: 1

2. View the exhibit:

```
public class Student {  
    public String name = "";  
    public int age = 0;  
    public String major = "Undeclared";  
    public boolean fulltime = true;  
    public void display() {  
        System.out.println("Name: " + name + " Major: " + major);  
    }  
    public boolean isFullTime() {  
        return fulltime;  
    }  
}
```

Given:

```
Public class TestStudent {  
    Public static void main(String[] args) {  
        Student bob = new Student ();  
        Student jian = new Student();  
        bob.name = "Bob";  
        bob.age = 19;  
        jian = bob; jian.name = "Jian";  
        System.out.println("Bob's Name: " + bob.name);  
    }  
}
```

What is the result when this program is executed?

- A. Bob's Name: Bob

- B.Bob's Name: Jian
- C.Nothing prints
- D.Bob's name

Answer: B

Explanation: After the statement `jian = bob;` the `jian` will reference the same object as `bob`.

3.Given the code fragment:

```
String valid = "true";  
if (valid) System.out.println ("valid");  
else system.out.println ("not valid");
```

What is the result?

- A.Valid
- B.not valid
- C.Compilation fails
- D.An `IllegalArgumentException` is thrown at run time

Answer: C

Explanation: In segment 'if (valid)' `valid` must be of type `boolean`, but it is a string. This makes the compilation fail.

4.Given:

```
public class ScopeTest {  
    int z;  
    public static void main(String[] args){  
        ScopeTest myScope = new ScopeTest();  
        int z = 6;  
        System.out.println(z);  
        myScope.doStuff();  
        System.out.println(z);  
        System.out.println(myScope.z);  
    }  
    void doStuff() {  
        int z = 5;  
        doStuff2();  
        System.out.println(z);  
    }  
    void doStuff2() {  
        z=4;  
    }  
}
```

What is the result?

- A.6 5 6 4
- B.6 5 5 4
- C.6 5 6 6
- D.6 5 6 5

Answer: A

Explanation: Within main z is assigned 6.z is printed.Output: 6

Within doStuff z is assigned 5.DoStuff2 locally sets z to 4 (but MyScope.z is set to 4), but in Dostuff z is still 5.z is printed.Output: 5

Again z is printed within main (with local z set to 6).Output: 6

Finally MyScope.z is printed.MyScope.z has been set to 4 within doStuff2().Output: 4

5.Which two are valid instantiations and initializations of a multi dimensional array?

A.int [] [] array 2D = { { 0, 1, 2, 4} {5, 6}};

B.int [] [] array2D = new int [2] [2];

array2D[0] [0] = 1;

array2D[0] [1] = 2;

array2D[1] [0] = 3;

array2D[1] [1] = 4;

C.int [] [] [] array3D = {{0, 1}, {2, 3}, {4, 5}};

D.int [] [] [] array3D = new int [2] [2] [2];

array3D [0] [0] = array;

array3D [0] [1] = array;

array3D [1] [0] = array;

array3D [0] [1] = array;

E.int [] [] array2D = {0, 1};

Answer: B,D

Explanation: In the Java programming language, a multidimensional array is simply an array whose components are themselves arrays.

6.An unchecked exception occurs in a method dosomething()

Should other code be added in the dosomething() method for it to compile and execute?

A.The Exception must be caught

B.The Exception must be declared to be thrown.

C.The Exception must be caught or declared to be thrown.

D.No other code needs to be added.

Answer: D

Explanation: Because the Java programming language does not require methods to catch or to specify unchecked exceptions (RuntimeException, Error, and their subclasses), programmers may be tempted to write code that throws only unchecked exceptions or to make all their exception subclasses inherit from RuntimeException.Both of these shortcuts allow programmers to write code without bothering with compiler errors and without bothering to specify or to catch any exceptions.Although this may seem convenient to the programmer, it sidesteps the intent of the catch or specify requirement and can cause problems for others using your classes.

7.Given the code fragment:

```
int b = 4;
```

```
b -- ;
```

```
System.out.println (-- b);
```

```
System.out.println(b);
```

What is the result?

A.2 2

B.1 2

C.3 2

D.3 3

Answer: A

Explanation: Variable b is set to 4.

Variable b is decreased to 3.

Variable b is decreased to 2 and then printed.Output: 2

Variable b is printed.Output: 2

8.Given the code fragment:

```
interface SampleClosable {  
    public void close () throws java.io.IOException;  
}
```

Which three implementations are valid?

A.public class Test implements SampleCloseable {

```
    public void close() throws java.io.IOException {
```

```
        // do something
```

```
    }
```

```
}
```

B.public class Test implements SampleCloseable {

```
    public void close() throws Exception {
```

```
        // do something
```

```
    }
```

```
}
```

C.public class Test implements SampleCloseable {

```
    public void close() throws java.io.FileNotFoundException {
```

```
        // do something
```

```
    }
```

```
}
```

D.public class Test extends SampleCloseable {

```
    public void close() throws java.io.IOException {
```

```
        // do something
```

```
    }
```

```
}
```

E.public class Test implements SampleCloseable {

```
    public void close()
```

```
        // do something
```

```
    }
```

```
}
```

Answer: A,C,E

Explanation: A: Throwing the same exception is fine.

C: Using a subclass of java.io.IOException (here java.io.FileNotFoundException) is fine

E: Not using a throw clause is fine.

9. Given the code fragment:

```
int [][] array = {{0}, {0, 1}, {0, 2, 4}, {0, 3, 6, 9}, {0, 4, 8, 12, 16}};
```

```
System.out.println(array [4] [1]);
```

```
System.out.println (array) [1][4];
```

```
int [][] array = {{0}, {0, 1}, {0, 2, 4}, {0, 3, 6, 9}, {0, 4, 8, 12, 16}};
```

```
System.out.println(array [4][1]);
```

```
System.out.println(array) [1][4];
```

What is the result?

A. 4 Null

B. Null 4

C. An IllegalArgumentException is thrown at run time

D. 4 An ArrayIndexOutOfBoundsException is thrown at run time

Answer: D

Explanation: The first println statement, System.out.println(array [4][1]);, works fine. It selects the element/array with index 4, {0, 4, 8, 12, 16}, and from this array it selects the element with index 1,

4. Output: 4

The second println statement, System.out.println(array) [1][4]);, fails. It selects the array/element with index 1, {0, 1}, and from this array it tries to select the element with index 4. This causes an exception.

Output: 4

Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 4

10. Given:

```
public class DoCompare1 {
```

```
public static void main(String[] args) {
```

```
String[] table = {"aa", "bb", "cc"};
```

```
for (String ss: table) {
```

```
int ii = 0;
```

```
while (ii < table.length) {
```

```
System.out.println(ss + ", " + ii);
```

```
ii++;
```

```
}
```

```
}
```

How many times is 2 printed as a part of the output?

A. Zero

B. Once

C. Twice

D. Thrice

E. Compilation fails.

Answer: D

Explanation: The for statement, for (String ss: table), is executed one time for each of the three

elements in table. The while loop will print a 2 once for each element.

Output:

aa, 0

aa, 1

aa, 2

bb, 0

bb, 1

bb, 2

cc, 0

cc, 1

cc, 2