



Vendor: Oracle

Exam Code: 1Z0-809

Exam Name: Java SE 8 Programmer II

Version: Demo

QUESTION 1

Which two statements are true for a two-dimensional array of primitive data type?

- A. It cannot contain elements of different types.
- B. The length of each dimension must be the same.
- C. At the declaration time, the number of elements of the array in each dimension must be specified.
- D. All methods of the class object may be invoked on the two-dimensional array.

Correct Answer: CD

Explanation:

<http://stackoverflow.com/questions/12806739/is-an-array-a-primitive-type-or-an-object-or-something-else-entirely>

QUESTION 2

Given the fragments:

```
public class TestA extends Root {
    public static void main(String[] args) {
        Root r = new TestA();
        System.out.println(r.method1()); // line n1
        System.out.println(r.method2()); // line n2
    }
}
class Root {
    private static final int MAX = 20000;
    private int method1() {
        int a = 100 + MAX; // line n3
        return a;
    }
    protected int method2() { // line n4
        int a = 200 + MAX;
        return a;
    }
}
```

Which line causes a compilation error?

- A. Line n1
- B. Line n2
- C. Line n3
- D. Line n4

Correct Answer: A

QUESTION 3

Given the code fragment:

```
List<String> listVal = Arrays.asList("Joe", "Paul", "Alice", "Tom");
```

```
System.out.println (
```

```
// line n1
```

```
);
```

Which code fragment, when inserted at line n1, enables the code to print the count of string elements whose length is greater than three?

- A. `listVal.stream().filter(x -> x.length()>3).count()`
- B. `listVal.stream().map(x -> x.length()>3).count()`
- C. `listVal.stream().peek(x -> x.length()>3).count().get()`
- D. `listVal.stream().filter(x -> x.length()>3).mapToInt(x -> x).count()`

Correct Answer: C

QUESTION 4

Given:

1. `abstract class Shape {`
2. `Shape () { System.out.println (摺hape?; }`
3. `protected void area () { System.out.println (摺hape?; }`
4. `}`
- 5.
6. `class Square extends Shape {`
7. `int side;`
8. `Square int side {`
9. `/* insert code here */`
10. `this.side = side;`
11. `}`
12. `public void area () { System.out.println (摺quare?; }`
13. `}`
14. `class Rectangle extends Square {`
15. `int len, br;`
16. `Rectangle (int x, int y) {`
17. `/* insert code here */`
18. `len = x, br = y;`
19. `}`

```
20. void area ( ) { System.out.println (擧ectangle?; }
```

```
21. }
```

Which two modifications enable the code to compile?

- A. At line 1, remove abstract
- B. At line 9, insert super ();
- C. At line 12, remove public
- D. At line 17, insert super (x);
- E. At line 17, insert super (); super.side = x;
- F. At line 20, use public void area () {

Correct Answer: CD

QUESTION 5

Which two are Java Exception classes?

- A. SecurityException
- B. DuplicatePathException
- C. IllegalArgumentException
- D. TooManyArgumentsException

Correct Answer: AC

QUESTION 6

Given the code fragments:

```
class TechName {
```

```
String techName;
```

```
TechName (String techName) {
```

```
this.techName=techName;
```

```
}
```

```
}
```

and

```
List<TechName> tech = Arrays.asList (
```

```
new TechName("Java-"),
```

```
new TechName("Oracle DB-"),
```

```
new TechName("J2EE-")
```

```
);
```

```
Stream<TechName> stre = tech.stream();  
  
//line n1
```

Which should be inserted at line n1 to print Java-Oracle DB-J2EE-?

- A. stre.forEach(System.out::print);
- B. stre.map(a-> a.techName).forEach(System.out::print);
- C. stre.map(a-> a).forEachOrdered(System.out::print);
- D. stre.forEachOrdered(System.out::print);

Correct Answer: C

QUESTION 7

Given the definition of the Vehicle class:

```
class Vehicle {  
  
String name;  
  
void setName (String name) {  
  
this.name = name;  
  
}  
  
String getName() {  
  
return name;  
  
}  
  
}
```

Which action encapsulates the Vehicle class?

- A. Make the Vehicle class public.
- B. Make the name variable public.
- C. Make the setName method public.
- D. Make the name variable private.
- E. Make the setName method private.
- F. Make the getName method private.

Correct Answer: B

QUESTION 8

Given the code fragment:

```
List<String> empDetails = Arrays.asList("100, Robin, HR",  
  
"200, Mary, AdminServices",
```

```
"101, Peter, HR");  
empDetails.stream()  
.filter(s-> s.contains("1"))  
.sorted()  
.forEach(System.out::println); //line n1
```

What is the result?

- A. 100, Robin, HR
101, Peter, HR
- B. A compilation error occurs at line n1.
- C. 100, Robin, HR
101, Peter, HR
200, Mary, AdminServices
- D. 100, Robin, HR
200, Mary, AdminServices
101, Peter, HR

Correct Answer: C

QUESTION 9

Given:

```
public class X {  
    public static void main(String[] args){  
        String theString = "Hello World";  
        System.out.println(theString.charAt(11));  
    }  
}
```

What is the result?

- A. The program prints nothing
- B. A NullPointerException is thrown at runtime.
- C. A StringIndexOutOfBoundsException is thrown at runtime.
- D. AnArrayIndexOutOfBoundsException is thrown at runtime.

Correct Answer: C

QUESTION 10

Given:

```
public class Foo<K, V> {  
  
    private K key;  
  
    private V value;
```

```
public Foo (K key, V value) (this.key = key; this.value = value;)  
public static <T> Foo<T, T> twice (T value) (return new Foo<T, T> (value, value); )  
public K getKey () (return key;)  
public V getValue () (return value;)  
}
```

Which option fails?

- A. Foo<String, Integer> mark = new Foo<String, Integer> ("Steve", 100);
- B. Foo<String, String> pair = Foo.<String>twice ("Hello World!");
- C. Foo<?, ?> percentage = new Foo <> (97, 32);
- D. Foo<String, String> grade = new Foo <> ("John", "A");

Correct Answer: C

QUESTION 11

Given:

```
public class product {  
    int id; int price;  
    public Product (int id, int price) {  
        this.id = id;  
        this.price = price;  
    }  
    public String toString() { return id + ":" + price; }  
}
```

and the code fragment:

```
List<Product> products = Arrays.asList(new Product(1, 10),  
    new Product (2, 30),  
    new Product (2, 30));  
Product p = products.stream().reduce(new Product (4, 0), (p1, p2) -> {  
    p1.price+=p2.price;  
    return new Product (p1.id, p1.price);});  
products.add(p);
```

```
products.stream().parallel()
.reduce((p1, p2) -> p1.price > p2.price ? p1 : p2)
.ifPresent(System.out::println);
```

What is the result?

- A. 2 : 30
- B. 4 : 0
- C. 4 : 60
- D. 4 : 60
2 : 30
3 : 20
1 : 10
- E. The program prints nothing.

Correct Answer: D

QUESTION 12

Given the code fragment:

```
List<Integer> values = Arrays.asList (1, 2, 3);
values.stream ()
.map(n -> n*2)//line n1
.peek(System.out::print)//line n2
.count();
```

What is the result?

- A. 246
- B. The code produces no output.
- C. A compilation error occurs at line n1.
- D. A compilation error occurs at line n2.

Correct Answer: A

QUESTION 13

Given the code fragments:


```
interface Contract{ }
class Super implements Contract{ }
class Sub extends Super {

public class Ref {
    public static void main(String[] args) {
        List objs = new ArrayList();

        Contract c1 = new Super();
        Contract c2 = new Sub(); // line n1
        Super s1 = new Sub();

        objs.add(c1);
        objs.add(c2); // line n2
        objs.add(s1);

        for(Object itm: objs) {
            System.out.println(itm.getClass().getName());
        }
    }
}
```

What is the result?

- A. Super
Sub
Sub
- B. Contract
Contract
Super
- C. Compilation fails at line n1
- D. Compilation fails at line n2

Correct Answer: D

QUESTION 14

Given the content of /resources/Message.properties:

```
welcome1="Good day!"
```

and given the code fragment:

```
Properties prop = new Properties ();
```

```
FileInputStream fis = new FileInputStream ("/resources/Message.properties");
```

```
prop.load(fis);
```

```
System.out.println(prop.getProperty("welcome1"));
```

```
System.out.println(prop.getProperty("welcome2", "Test")); //line n1
```

```
System.out.println(prop.getProperty("welcome3"));
```

What is the result?

- A. Good day!

- Test
followed by an Exception stack trace
- B. Good day!
followed by an Exception stack trace
- C. Good day!
Test
null
- D. A compilation error occurs at line n1.

Correct Answer: D

QUESTION 15

Given:

```
Class A { }
```

```
Class B { }
```

```
Interface X { }
```

```
Interface Y { }
```

Which two definitions of class C are valid?

- A. Class C extends A implements X { }
- B. Class C implements Y extends B { }
- C. Class C extends A, B { }
- D. Class C implements X, Y extends B { }
- E. Class C extends B implements X, Y { }

Correct Answer: AE

Explanation:

extends is for extending a class.

implements is for implementing an interface.

Java allows for a class to implement many interfaces.

QUESTION 16

Which statement is true about java.util.stream.Stream?

- A. A stream cannot be consumed more than once.
- B. The execution mode of streams can be changed during processing.
- C. Streams are intended to modify the source data.
- D. A parallel stream is always faster than an equivalent sequential stream.

Correct Answer: B