



Oracle

Exam Questions 1z0-808

Java SE 8 Programmer I

NEW QUESTION 1

Given the code fragments:

Person.java:

```
public class Person {
    String name;
    int age;

    public Person(String n, int a) {
        name = n;
        age = a;
    }

    public String getName() {
        return name;
    }

    public int getAge() {
        return age;
    }
}
```

Test.java:

```
public static void checkAge(List<Person> list, Predicate<Person> predicate) {
    for (Person p : list) {
        if (predicate.test(p)) {
            System.out.println(p.name + " ");
        }
    }
}

public static void main(String[] args) {
    List<Person> iList = Arrays.asList(new Person("Hank", 45),
                                       new Person("Charlie", 40),
                                       new Person("Smith", 38));

    //line n1
}
```

Which code fragment, when inserted at line n1, enables the code to print Hank?

- A `checkAge (iList, () -> p. get Age () > 40);`
- B `checkAge(iList, Person p -> p.getAge() > 40);`
- C `checkAge (iList, p -> p.getAge () > 40);`
- D `checkAge(iList, (Person p) -> { p.getAge() > 40; });`

- A. Option A
- B. Option B
- C. Option C
- D. Option D

Answer: C

NEW QUESTION 2

You are asked to develop a program for a shopping application, and you are given this information:

- > The application must contain the classes Toy, EduToy, and ConsToy. The Toy class is the superclass of the other two classes.
- > The int calculatePrice (Toy t) method calculates the price of a toy.
- > The void printToy (Toy t) method prints the details of a toy.

Which definition of the Toy class adds a valid layer of abstraction to the class hierarchy?

- A**
- ```
public abstract class Toy{
 public abstract int calculatePrice(Toy t);
 public void printToy(Toy t) { /* code goes here */ }
}
```
- B**
- ```
public abstract class Toy {
    public int calculatePrice(Toy t) ;
    public void printToy(Toy t) ;
}
```
- C**
- ```
public abstract class Toy {
 public int calculatePrice(Toy t);
 public final void printToy(Toy t){ /* code goes here */ }
}
```
- D**
- ```
public abstract class Toy {
    public abstract int calculatePrice(Toy t) { /* code goes here */ }
    public abstract void printToy(Toy t) { /* code goes here */ }
}
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D

Answer: A

NEW QUESTION 3

Given the definitions of the MyString class and the Test class:

```
package p1;
class MyString {
    String msg;
    MyString(String msg) {
        this.msg = msg;
    }
}
```

Test.java:

```
package p1;
public class Test {
    public static void main(String[] args) {
        System.out.println("Hello " + new StringBuilder("Java SE 8"));
        System.out.println("Hello " + new MyString("Java SE 8").msg);
    }
}
```

What is the result?

- A**
- ```
Hello Java SE 8
Hello Java SE 8
```
- B**
- ```
Hello java.lang.StringBuilder@<<hashCode1>>
Hello p1.MyString@<<hashCode2>>
```
- C**
- ```
Hello Java SE 8
Hello p1.MyString@<<hashCode>>
```
- D** Compilation fails at the Test class

- A. Option A
- B. Option B

- C. Option C
- D. Option D
- E. Option E

**Answer: D**

#### NEW QUESTION 4

Given the code fragment:

```
LocalDate Time dt= LocalDateTime.of (2014, 7, 31, 1, 1);
dt.plusDays (30);
dt. plusMonths (1);
System.out.print (dt format (DateTimeFormatter. ISO_DATE));
```

What is the result?

- A. An exception is thrown at runtime
- B. 07-31-2014
- C. 2014-07-31
- D. 2014-09-30

**Answer: A**

#### NEW QUESTION 5

Given the code fragment:

```
public static void main(String[] args) {
 short s1 = 200;
 Integer s2 = 400;
 Long s3 = (long) s1 + s2; //line n1
 String s4 = (String) (s3 * s2); //line n2
 System.out.println("Sum is " + s4);
}
```

What is the result?

- A. Sum is 600
- B. Compilation fails at line n1.
- C. Compilation fails at line n2.
- D. A ClassCastException is thrown at line n1.
- E. A ClassCastException is thrown at line n2.

**Answer: C**

#### NEW QUESTION 6

Given the code fragment:

```
public static void main(String[] args) {
 int data[] = {2010, 2013, 2014, 2015, 2014};
 int key = 2014;
 int count = 0;
 for (int e: data) {
 if (e != key) {
 continue;
 count++;
 }
 }
 System.out.print(count + " Found");
}
```

What is the result?

- A. Compilation fails.
- B. 0 Found
- C. 1 Found
- D. 3 Found

**Answer: A**

#### NEW QUESTION 7

Given the code fragment:

```
int n [] [] = {{1, 3}, {2, 4}};
for (int i = n.length-1; i >= 0; i--) {
 for (int y : n[i]) {
 System.out.print (y);
 }
}
```

What is the result?

- A. 1324
- B. 2313
- C. 3142
- D. 4231

**Answer: D**

#### NEW QUESTION 8

Given the code fragment:

```
public static void main(String[] args) {
 int ii = 0;
 int jj = 7;
 for (ii = 0; ii < jj - 1; ii = ii + 2) {
 System.out.print(ii + " ");
 }
}
```

What is the result?

- A. 2 4
- B. 0 2 4 6
- C. 0 2 4
- D. Compilation fails

**Answer: C**

#### NEW QUESTION 9

Given:

```
public class App {
 int count;
 public static void displayMsg() {
 System.out.println("Welcome Visit Count: " + count++); // line n1
 }
 public static void main(String[] args) {
 App.displayMsg();
 displayMsg(); // line n2
 }
}
```

What is the result?

- A. Welcome Visit Count:0Welcome Visit Count: 1
- B. Compilation fails at line n2.
- C. Compilation fails at line n1.
- D. Welcome Visit Count:0Welcome Visit Count: 0

**Answer: C**

**Explanation:**

```
1
2 public class App {
3 int count;
4 public static void displayMsg() {
5 System.out.println("Welcome Visit Count: " + count ++); //line n1
6 }
7 public static void main(String[] args) {
8 App.displayMsg();
9 displayMsg();
10 }
11 }
12
```

#### NEW QUESTION 10

Given these two classes:

```
public class Customer {
 ElectricAccount acct = new ElectricAccount ();

 public void useElectricity(double kWh) {
 acct.addKWh(kWh);
 }
}

public class ElectricAccount {
 private double kWh;
 private double rate = 0.07;
 private double bill;

 //line n1
}
```

Any amount of electricity used by a customer (represented by an instance of the Customer class) must contribute to the customer's bill (represented by the member variable bill) through the useElectricity method.

An instance of the Customer class should never be able to tamper with or decrease the value of the member variable bill.

How should you write methods in the ElectricAccount class at line n1 so that the member variable bill is always equal to the value of the member variable kWh multiplied by the member variable rate?

A

```
public void addKWh(double kWh) {
 this.kWh += kWh;
 this.bill = this.kWh*this.rate;
}
```

B

```
public void addKWh(double kWh) {
 if (kWh > 0) {
 this.kWh += kWh;
 this.bill = this.kWh * this.rate;
 }
}
```

C

```
private void addKWh(double kWh) {
 if (kWh > 0) {
 this.kWh += kWh;
 this.bill = this.kWh*this.rate;
 }
}
```

D

```
public void addKWh(double kWh) {
 if(kWh > 0) {
 this.kWh += kWh;
 setBill(this.kWh);
 }
}

public void setBill(double kWh) {
 bill = kWh*rate;
}
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D

Answer: A

#### NEW QUESTION 10

Which two statements are true about Java byte code? (Choose two.)

- A. It can be serialized across network.
- B. It can run on any platform that has a Java compiler.
- C. It can run on any platform.
- D. It has ".java" extension.
- E. It can run on any platform that has the Java Runtime Environment.

Answer: AE

**NEW QUESTION 15**

This grid shows the state of a 2D array:

|   |   |   |
|---|---|---|
| 0 | 0 |   |
|   | X | 0 |
| X |   | X |

The grid is created with this code:

```
char[][] grid = new char[3][3];
grid[1][1] = 'X';
grid[0][0] = '0';
grid[2][0] = 'X';
grid[0][1] = '0';
grid[2][2] = 'X';
grid[1][2] = '0';
//line n1
```

Which line of code, when inserted in place of //line n1, adds an X into the grid so that the grid contains three consecutive Xs?

- A. grid[2][1] = 'X';
- B. grid[3][2] = 'X';
- C. grid[3][1] = 'X';
- D. grid[2][3] = 'X';

**Answer: D**

**NEW QUESTION 17**

Given:

```
class Patient {
 String name;
 public Patient (String name) {
 this.name = name;
 }
}
```

And the code fragment:

```
8. public class Test {
9. public static void main (String [] args) {
10. List ps = new ArrayList ();
11. Patient p2 = new Patient ("Mike);
12. ps.add(p2);
13.
14. // insert code here
15.
16. if (f >= 0) {
17. System.out.print ("Mike Found");
18. }
19. }
20. }
```

Which code fragment, when inserted at line 14, enables the code to print Mike Found?

- A  
`int f = ps.indexOf (p2);`
- B  
`int f = ps.indexOf (Patient ("Mike" ) );`
- C  
`int f = ps.indexOf (new Patient "Mike" ) ;`
- D  
`Patient p = new Patient ("Mike");`  
`int f = ps.indexOf(p)`

- A. Option A  
 B. Option B  
 C. Option C  
 D. Option D

**Answer: A**

### NEW QUESTION 22

Given:

```
class A {
 public void test() {
 System.out.println("A ");
 }
}

class B extends A {
 public void test() {
 System.out.println("B ");
 }
}

public class C extends A {
 public void test() {
 System.out.println("C ");
 }

 public static void main(String[] args) {
 A b1 = new A();
 A b2 = new C();
 A b3 = (B) b2; //line n1
 b1 = (A) b2; //line n2
 b1.test();
 b3.test();
 }
}
```

What is the result?

- A. AB  
 B. AC  
 C. CC  
 D. A ClassCastException is thrown only at line n1.  
 E. A ClassCastException is thrown only at line n2.

**Answer: D**

### NEW QUESTION 25

Given the code fragment:

```
public static void main(String[] args) {
 LocalDate date = LocalDate.of(2012, 01, 32);
 date.plusDays(10);
 System.out.println(date);
}
```

What is the result?

- A. 2012-02-10  
 B. 2012-02-11

- C. Compilation fails
- D. A DateTimeException is thrown at runtime.

**Answer:** D

**NEW QUESTION 27**

Given:

```
interface I {
 public void displayI();
}
abstract class C2 implements I {
 public void displayC2() {
 System.out.print("C2");
 }
}
class C1 extends C2 {
 public void displayI() {
 System.out.print("C1");
 }
}
```

And the code fragment:

```
C2 obj1 = new C1();
I obj2 = new C1();

C2 s = (C2) obj2;
I t = obj1;

t.displayI();
s.displayC2();
```

What is the result?

- A. C1C2
- B. C1C1
- C. Compilation fails.
- D. C2C2

**Answer:** A

**Explanation:**

lund

- src

App.java

```

1
2 interface I {
3 public void displayI();
4 }
5 abstract class C2 implements I {
6 public void displayC2() {
7 System.out.print("C2");
8 }
9 }
10 class C1 extends C2 {
11 public void displayI() {
12 System.out.print("C1");
13 }
14
15 }
16
17 public class App {
18 public static void main(String[] args) {
19 C2 obj1 = new C1();
20 I obj2 = new C1();
21
22 C2 s = (C2) obj2;
23 I t = obj1;
24
25 t.displayI();
26 s.displayC2();
27 }
28
29 }

```

Console 1 x Console 2 x Console 3 x Console 4 x

Console 1  
C1C2  
Completed with exit code: 0

**NEW QUESTION 28**

Given the code fragment:

```

public static void main(String[] args) {
 LocalDate date = LocalDate.of(2012, 1, 30);
 date.plusDays(10);
 System.out.println(date);
}

```

What is the result?

- A. 2012-02-10
- B. 2012-01-30
- C. 2012-02-10 00:00
- D. A DateTimeException is thrown at runtime.

**Answer: C**

**NEW QUESTION 31**

Given the code fragment:

```
public static void main(String[] args) {
 String myStr = "Hello World ";
 myStr.trim();
 int i1 = myStr.indexOf(" ");
 System.out.println(i1);
}
```

What is the result?

- A. An exception is thrown at runtime.
- B. -1
- C. 5
- D. 10

**Answer: A**

**NEW QUESTION 32**

Given the code fragment:

```
int wd = 0;
String days[] = {"sun", "mon", "wed", "sat"};
for (String s:days) {
 switch (s) {
 case "sat":
 case "sun":
 wd -= 1;
 break;
 case "mon":
 wd++;
 case "wed":
 wd += 2;
 }
}
System.out.println(wd);
```

What is the result?

- A. 3
- B. 4
- C. -1
- D. Compilation fails.

**Answer: A**

**NEW QUESTION 35**

Given:

```
public class Test {
 public static void main(String[] args) {
 Test ts = new Test();
 System.out.print(isAvailable + " ");
 isAvailable= ts.doStuff();
 System.out.println(isAvailable);
 }
 public static boolean doStuff() {
 return !isAvailable;
 }
 static boolean isAvailable = false;
}
```

What is the result?

- A. Compilation fails.
- B. false true
- C. true false
- D. true true
- E. false false

Answer: B

#### NEW QUESTION 37

Given:

```
class Student {
 String name;
 public Student(String name) {
 this.name = name;
 }
}

public class Test {
 public static void main(String[] args) {
 Student[] students = new Student[3];
 students[1] = new Student("Richard");
 students[2] = new Student("Donald");
 for (Student s : students) {
 System.out.println("" + s.name);
 }
 }
}
```

What is the result?

- A. nullRichardDonald
- B. RichardDonald
- C. Compilation fails.
- D. An ArrayIndexOutOfBoundsException is thrown at runtime.
- E. A NullPointerException is thrown at runtime.

Answer: E

#### NEW QUESTION 40

Given this class:

```
public class Rectangle {
 private double length;
 private double height;
 private double area;

 public void setLength(double length) {
 this.length = length;
 }
 public void setHeight(double height) {
 this.height = height;
 }
 public void setArea() {
 area = length*height;
 }
}
```

Which two changes would encapsulate this class and ensure that the area field is always equal to length \* height whenever the Rectangle class is used?

- A. Call the setArea method at the end of the setHeight method.
- B. Call the setArea method at the beginning of the setHeight method.
- C. Call the setArea method at the end of the setLength method.
- D. Call the setArea method at the beginning of the setLength method.
- E. Change the setArea method to private.
- F. Change the area field to public.

Answer: AE

#### NEW QUESTION 44

Given this segment of code:

```
ArrayList<Cycle> myList = new ArrayList<>();
myList.add(new Motorcycle());
```

Which two statements, if either were true, would make the code compile? (Choose two.)

- A. Motorcycle is an interface that implements the Cycle class.
- B. Cycle is an interface that is implemented by the Motorcycle class.
- C. Cycle is an abstract superclass of Motorcycle.
- D. Cycle and Motorcycle both extend the Transportation superclass.
- E. Cycle and Motorcycle both implement the Transportation interface.
- F. Motorcycle is a superclass of Cycle.

**Answer:** BC

**NEW QUESTION 47**

Which two statements are true? (Choose two.)

- A. Error class is unextendable.
- B. Error class is extendable.
- C. Error is a RuntimeException.
- D. Error is an Exception.
- E. Error is a Throwable.

**Answer:** BC

**NEW QUESTION 52**

Which three statements describe the object-oriented features of the Java language? (Choose three.)

- A. Objects cannot be reused.
- B. A subclass must override the methods from a superclass.
- C. Objects can share behaviors with other objects.
- D. A package must contain a main class.
- E. Object is the root class of all other objects.
- F. A main method must be declared in every class.

**Answer:** BCF

**NEW QUESTION 57**

Which statement will empty the contents of a StringBuilder variable named sb?

- A. s
- B. deleteAll ();
- C. s
- D. delete (0, s
- E. size () );
- F. s
- G. delete (0, s
- H. length () );
- I. s
- J. removeAll ();

**Answer:** C

**NEW QUESTION 60**

.....

## Thank You for Trying Our Product

### We offer two products:

1st - We have Practice Tests Software with Actual Exam Questions

2nd - Questions and Answers in PDF Format

### 1z0-808 Practice Exam Features:

- \* 1z0-808 Questions and Answers Updated Frequently
- \* 1z0-808 Practice Questions Verified by Expert Senior Certified Staff
- \* 1z0-808 Most Realistic Questions that Guarantee you a Pass on Your FirstTry
- \* 1z0-808 Practice Test Questions in Multiple Choice Formats and Updatesfor 1 Year

**100% Actual & Verified — Instant Download, Please Click**  
**[Order The 1z0-808 Practice Test Here](#)**