

# Object-Oriented Programming in Java

## Quiz 1

Jan 10, 2001

---

### Problem 1: Who wants to be a Java developer? (with apologies to Regis)

Fill in your answer in the space provided.

**Question 1:** Which of these word-pairs represents the IS-A relationship (in English)?

- A. duck/bird
- B. bird/fly
- C. beak/bird
- D. bird/nest

Final Answer:

**Question 2:** Which of these pairs represents the HAS-A relationship?

- A. bird/duck
- B. bird/fly
- C. bird/beak
- D. bird/flying\_animal

Final Answer:

**Question 3:** Which of these pairs represents the BEHAVES-LIKE relationship?

- A. bird/duck
- B. bird/fly
- C. bird/beak
- D. bird/flying\_animal

Final Answer:

**Question 4:** Which phrase below best describes the relationship between Country and Capital?

- A. IS-A
- B. HAS-A
- C. USES
- D. BEHAVES-LIKE

Final Answer:

**Question 5:** Which Java technology best supports the IS-A relationship?

- A. Inheritance
- B. Access Restriction
- C. Strong Typing
- D. Garbage Collection

Final Answer:

**Question 6:** Which Java keyword is used to specify inheritance?

- A. extends
- B. implements
- C. public
- D. static

Final Answer:

**Question 7:** Which Java keyword is used to specify compliance with an interface?

- A. extends    B. implements
- C. public     D. static

Final Answer:

**Question 8:** Which Java keywords can not appear in a class declaration?

- A. extends    B. private
- C. void        D. abstract

Final Answer:

**Question 9:** Is it possible to use the techniques of object oriented programming in a language that does not specifically support OOP?

- A. Yes    B. No
- C.        D.

Final Answer:

**Question 10:** Are private classes only for malicious corporations closely guarding their proprietary code?

- A. Yes    B. No
- C.        D.

Final Answer:

**Question 11:** Which Java keywords can not appear in instance variable definitions?

- A. extends    B. protected
- C. int         D. private

Final Answer:

**Question 12:** Which Java keywords can not appear in method definitions?

- A. implements    B. void
- C. static            D. private

Final Answer:

**Question 13:** Which of these fragments represents the HAS-A relationship between Foo and Bar?

- A. class Foo extends Bar{ }                    B. class Foo implements Bar{ }
- C. class Foo { private Bar mybar; }        D. abstract class Bar extends Foo{ }

Final Answer:

**Question 14:** Which line of code in the following can be inserted in place of the comments, to perform the initialization described by the comments?

```
public class T {
    int r;
    int s;

    T(int x, int y) {
        r = x;
        s = y;
    }
}

class S extends T {
    int t;

    public S(int x, int y, int z){
        // insert here the code
        // that would do the correct initialization
        // r= x and s= y
        t=z;
    }
}
```

- A. T(x, y);      B. this(x, y);  
C. super(x, y);    D. super(x, y, z);

Final Answer:

**Question 15:** What do the 'public' and 'private' keywords relate to?

- A. Typing              B. Garbage Collection  
C. Polymorphism    D. Access Restriction

Final Answer:

**Question 16:** Which Java technologies most powerfully supports the programming concept of “encapsulation”?

- A. Container Classes    B. Access Restriction  
C. Garbage Collection    D. Typing

Final Answer:

**Question 17:** Which Java technologies most powerfully support the programming concept of “abstraction”?

- A. Inheritance and Polymorphism    B. Heap Allocation  
C. Garbage Collection                  D. Applets

Final Answer:

## Problem 2: Program Analysis: Short Answer

**Question 1:** What are the values of i,a, and b after the following code fragment runs.?

```
int i;
int a=1;
int b=0;
for(i=0;i<10;i++){
    a = a * 2;
    b++;
}
```

i =

a =

b =

**Question 2:** What is the value of a after the following code fragment runs?

```
int a=0;
while(true){
    if(a > 20) break;
    a = a+1;
}
```

a =

**Question 3:** What three lines will the following code print out when compiled and run?

```
public class Spindle{
    public static void main(String[] args){
        int[] a = new int[] {1,2,3,4};
        int b = 5;
        int c =6;

        Fold.mutilate( a , b, c);
        System.out.println( a[0]);
        System.out.println( b);
        System.out.println( c );
    }
}

class Fold{
    static void mutilate( int[] a, int b , int c) {
        a[0] = 7;
        b = 8;
        c = 9;
    }
}
```

line1:

line2:

line3:

**Question 4:** What does the class Test below print out if it is compiled and run?

```
public class Test
{
    int x;

    Test(String s){
        this();
        System.out.println(s);
        System.out.println("This");
    }

    Test(int x){
        this("is");
    }

    Test(){
        System.out.println("a test.");
    }

    public static void main(String[] args)
    {
        int val = 2;
        Test t = new Test(val);
    }
}
```

Test prints out:

The next three short answer question use the following group of classes.

```
/**
 * An interface with one method
 */
interface Face1{
    public void bar();
}

/**
 * A class with an accessor and method foo()
 */
class Class1{
    private int size = 0;
    private String name;

    Class1(String name){
        this.name = name;
    }

    public String getName(){
        return(name);
    }

    public void foo(){
        System.out.println('Class1.foo()');
    }
}

/**
 * A class inheriting from Class1
 */
class Class2 extends Class1 implements Face1{
    int size = 0;
    int x;

    Class2(String name){
        super(name);
    }

    public void foo(){
        System.out.println('Class2.foo()');
    }

    public void bar(){
        System.out.println('Class2.bar()');
    }
}
```

**Question 5:** What does the following print out?

```
public class TestClass{

    public static void main(String[] args){
        Class1 c = new Class2('me');
        c.foo();
    }

}
```

Ans:

**Question 6:** What does the following print out?

```
public class TestClass{

    public static void main(String[] args){
        Class1 c = new Class1('me');
        c.foo();
    }

}
```

Ans:

**Question 7:** What does the following print out?

```
public class TestClass{

    public static void main(String[] args){
        Class2 c = new Class2('me');
        System.out.println(c.getName());
    }

}
```

Ans:

**Question 8:** What does the following print out?

```
public class TestClass{

    public static void main(String[] args){
        Face1 c = new Class2('me');
        c.bar();
    }

}
```

Ans:

### Problem 3: Class Implementation

In this problem we ask you to implement part of a class Fraction, which represents fractions of the form  $a/b$  where  $a$  and  $b$  are integers, (1/2, 4/6, 101/432. etc.). In the space below, complete the class Fraction. Use `int` to store the numerator and denominator.

Include a constructor that initializes from a numerator and denominator.

Write an instance method for addition. [Hint:  $a/b + c/d = (ad + bc)/bd$ .]

Write a static method for multiplication. [Hint:  $a/b * c/d = ac/bd$ .]

Write a static `main()` that allocates two Fractions, 1/2 and 3/4 and stores their sum in a third variable.

(Be merciful to graders: please write clearly).

```
/**
 * Class Fraction. Represent the fraction X/Y where X and Y are integers
 */
public class Fraction{
    /**
     * Your instance vars here
     */
```

```
    /**
     * Your constructor here
     */
```

Continued on next page

```
/**  
 * Your add method here  
 */
```

```
/**  
 * Your static mult method here  
 */
```

```
/**  
 * Your static main here  
 */
```

```
}
```

## Problem 4: OOP Design

You are designing an OO program to help people plan an evening out. It is proposed that the following classes be used. Add inheritance ( **extends** phrases ) and instance variables to show the IS-A and HAS-A relationships between these classes. Do not introduce any additional classes or and instance variables of any basic type. [Note: By Business, we mean anything that would appear in the Yellow Pages].

```
class Address
{

}

class Business
{

}

class City
{

}

class Restaurant
{

}

class Theater
{

}
```