

Java Software Development Quiz 8

1. Which statements are true?

Select the two correct answers.

- (a) In Java, the extends clause is used to specify the inheritance relationship.
- (b) The subclass of a non-abstract class can be declared abstract.
- (c) All members of the superclass are inherited by the subclass.
- (d) A final class can be abstract.
- (e) A class in which all the members are declared private, cannot be declared public.

2. Which statements are true?

Select the two correct answers.

- (a) A class can only be extended by one class.
- (b) Every Java object has a public method named equals.
- (c) Every Java object has a public method named length.
- (d) A class can extend any number of classes.
- (e) A non-final class can be extended by any number of classes.

3. Given the following classes and declarations, which statements are true?

```
// Classes
class Foo {
    private int i;
    public void f() { /* ... */ }
    public void g() { /* ... */ }
}
class Bar extends Foo {
    public int j;
    public void g() { /* ... */ }
}
// Declarations:
Foo a = new Foo();
Bar b = new Bar();
```

Select the three correct answers.

- (a) The Bar class is a subclass of Foo.
- (b) The statement b.f(); is legal.
- (c) The statement a.j = 5; is legal.
- (d) The statement a.g(); is legal.
- (e) The statement b.i = 3; is legal.

4. Given classes A, B, and C, where B extends A, and C extends B, and where all classes implement the instance method void doIt(). How can the doIt() method in A be called from an instance method in C?

Select the one correct answer.

- (a) doIt();
- (b) super.doIt();
- (c) super.super.doIt();
- (d) this.super.doIt();
- (e) A.this.doIt();
- (f) ((A) this).doIt();
- (g) It is not possible.

5. Which method declarations, when inserted at (7), will not result in a compile-time error?

```
class MySuperclass {  
    public Integer step1(int i) { return 1; } // (1)  
    protected String step2(String str1, String str2) { return str1; } // (2)  
    public String step2(String str1) { return str1; } // (3)  
    public static String step2() { return "Hi"; } // (4)  
    public MyClass makeIt() { return new MyClass(); } // (5)  
    public MySuperclass makeIt2() { return new MyClass(); } // (6)  
}  
public class MyClass extends MySuperclass {  
    // (7) INSERT METHOD DECLARATION HERE  
}
```

Select the two correct answers.

- (a) public int step1(int i) { return 1; }
- (b) public String step2(String str2, String str1) { return str1; }
- (c) private void step2() { }
- (d) private static void step2() { }
- (e) private static String step2(String str) { return str; }
- (f) public MySuperclass makeIt() { return new MySuperclass(); }
- (g) public MyClass makeIt2() { return new MyClass(); }

6. What would be the result of compiling and running the following program?

```
class Vehicle {
    static public String getModelName() {
        return "Volvo";
    }

    public long getRegNo() {
        return 12345;
    }
}

class Car extends Vehicle {
    static public String getModelName() {
        return "Toyota";
    }

    public long getRegNo() {
        return 54321;
    }
}

public class TakeARide {
    public static void main(String args[]) {
        Car c = new Car();
        Vehicle v = c;
        System.out.println("|" + v.getModelName() + "|" + c.getModelName()
            + "|" + v.getRegNo() + "|" + c.getRegNo() + "|");
    }
}
```

Select the one correct answer.

- (a) The code will fail to compile.
- (b) The code will compile and print |Toyota|Volvo|12345|54321|, when run.
- (c) The code will compile and print |Volvo|Toyota|12345|54321|, when run.
- (d) The code will compile and print |Toyota|Toyota|12345|12345|, when run.
- (e) The code will compile and print |Volvo|Volvo|12345|54321|, when run.
- (f) The code will compile and print |Toyota|Toyota|12345|12345|, when run.
- (g) The code will compile and print |Volvo|Toyota|54321|54321|, when run.

Answer

1. (a) and (b)

The extends clause is used to specify that a class extends another class. A subclass can be declared abstract regardless of whether the superclass was declared abstract. Private, overridden, and hidden members from the superclass are not inherited by the subclass. A class cannot be declared both abstract and final, since an abstract class needs to be extended to be useful, and a final class cannot be extended. The accessibility of the class is not limited by the accessibility of its members. A class with all the members declared private can still be declared public.

2. (b) and (e)

The Object class has a public method named equals, but it does not have any method named length. Since all classes are subclasses of the Object class, they all inherit the equals() method. Thus, all Java objects have a public method named equals. In Java, a class can only extend a single superclass, but there is no limit on how many classes can extend a superclass.

3. (a), (b), and (d)

Bar is a subclass of Foo that overrides the method g(). The statement a.j = 5 is not legal, since the member j in the class Bar cannot be accessed through a Foo reference. The statement b.i = 3 is not legal either, since the private member i cannot be accessed from outside of the class Foo.

4. (g)

It is not possible to invoke the doIt() method in A from an instance method in class C. The method in C needs to call a method in a superclass two levels up in the inheritance hierarchy. The super.super.doIt() strategy will not work, since super is a keyword and cannot be used as an ordinary reference, nor accessed like a field. If the member to be accessed had been a field, the solution would be to cast the this reference to the class of the field and use the resulting reference to access the field. Field access is determined by the declared type of the reference, whereas the instance method to execute is determined by the actual type of the object denoted by the reference.

5. (b) and (g)

(a) and (1) do not have covariant return types. (b) overrides (2). The instance method in (c) cannot override the static method at (4). The static method in (d) and the static method at (4) do not have compatible return types. The static method in (e) cannot override the instance method at (3). The instance method in (f) and the instance method at (5) do not have compatible return types. The instance method in (g) overrides the instance method at (6), and they have covariant return types.

6. (g)

In the class Car, the static method getModelName() hides the static method of the same name in the superclass Vehicle. In the class Car, the instance method getRegNo() overrides the instance method of the same name in the superclass Vehicle. The declared type of the reference determines the method to execute when a static method is called, but the actual type of the object at runtime determines the method to execute when an overridden method is called.