

## Java 111 Chapter 8 (from version 1)

Total Questions: 17

Most Correct Answers: #1

Least Correct Answers: #3

1. An abstract class can only have abstract methods.

1/4 ☐ A True

2/4 ☒ B False

2. When you don't want a class to be instantiated (in other words, you don't want anyone to make a new object of that class type) mark the class with the "abstract" keyword.

2/4 ☒ A True

0/4 ☐ B False

3. Which of the following are true?

1/4 ☐ A An interface must be created using the keyword "abstract".

1/4 ☒ B An interface defines only abstract methods.

1/4 ☒ C A class can implement multiple interfaces.

2/4 ☒ D All interface methods are implicitly public.

4. All objects come out of an `ArrayList<Object>` as type `Object`, unless you use a cast.

2/4 ☒ A True

0/4 ☐ B False

5. Multiple inheritance is allowed in Java, meaning you may extend multiple classes.

0/4 ☐ A True

2/4 ☒ B False

6. If you override a superclass method in a subclass, you cannot invoke (call) the superclass method.

0/4 ☐ A True

2/4 ☒ B False

7. You can extend only one class (i.e. you can have only one immediate superclass).

2/4 ☒ A True

0/4 ☐ B False

8. Write an abstract method called eatCake that accepts one parameter for the number of slices to eat and returns a String.

Anon anon3889b6418b404cfe

✗ public abstract String eatCake(int numberOfSlices) {};

Anon anonf0b3f17f62b34f37

✗ public String eatCake(int slices);

9. Given the following: `JavaRockStar rockstar = new JavaRockStar();` what is the object reference variable?

1/4 ☐ A JavaRockStar

0/4 ☐ B new

1/4 ☒ C rockstar

0/4 ☐ D none of the above

10. Given the following: `JavaRockStar rockstar = new JavaRockStar();` what is the object reference type?

2/4 ☒ A JavaRockStar

0/4 ☐ B new

0/4 ☐ C rockstar

0/4 ☐ D none of the above

11. Given the following: `SuperStarCoder rockstar = new JavaRockStar();` what is the object reference type?

0/4 ☐ A JavaRockStar

0/4 ☐ B rockstar

2/4 ☒ C SuperStarCoder

0/4 ☐ D new

0/4 ☐ E none of the above

12. Given the following: SuperStarCoder rockstar = new JavaRockStar(); what is the actual object type?

- 2/4 ☒ A JavaRockStar
- 0/4 ☐ B rockstar
- 0/4 ☐ C SuperStarCoder
- 0/4 ☐ D new
- 0/4 ☐ E none of the above

13. Given the following, what output do you expect?

- 0/4 ☐ A Line 10 and 14 will each run twice.
- 0/4 ☐ B Line 10 will run twice, line 14 will run once.
- 0/4 ☒ C This will not compile due to line 29.
- 1/4 ☐ D This will not compile due to line 27.
- 0/4 ☐ E None of the above

```
1 public interface Programmer {
2     public abstract void writeCode();
3 }
4
5 public class SuperStarCoder implements Programmer {
6     public void writeCode() {
7         System.out.println("I'm a SuperStarCoder and I'm writing code!");
8     }
9 }
10
11 public class JavaRockStar implements Programmer {
12     public void writeCode() {
13         System.out.println("I'm a JavaRockStar and I'm writing code!");
14     }
15 }
16
17 public class ProgrammerTester {
18     public static void main(String[] args) {
19         Programmer p1 = new SuperStarCoder();
20         Programmer p2 = new JavaRockStar();
21         Programmer p3 = new Programmer();
22         p1.writeCode();
23         p2.writeCode();
24         p3.writeCode();
25     }
26 }
27
28 // Line 29: This line will cause a compilation error.
29 Programmer p4 = new Programmer();
30 p4.writeCode();
```

14. What is the proper way to create an interface called Payable?

- 1/4 ☐ A public abstract interface class Payable {}
- 0/4 ☐ B public abstract Payable {}
- 0/4 ☐ C public abstract interface class Payable extends Payable {}
- 0/4 ☒ D public interface Payable {}
- 0/4 ☐ E interface PayMe() extends Money implements Payable()

15. A class must extend a superclass before it can implement an interface.

- 0/4 ☐ A True
- 1/4 ☒ B False

16. If a class does not pass the IS-A test, it probably should not extend anything (other than Object).

- 1/4 ☒ A True
- 0/4 ☐ B False

17. An interface is a 100% abstract class, meaning it defines only abstract methods.

1/4



True

0/4



False