

Java 111 Chapter 8 (from version 1)

Total Questions: 17

Most Correct Answers: #10

Least Correct Answers: #8

1. An abstract class can only have abstract methods.

0/3 ☐ A True

3/3 ☒ 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.

3/3 ☒ A True

0/3 ☐ B False

3. Which of the following are true?

0/3 ☐ A An interface must be created using the keyword "abstract".

2/3 ☒ B An interface defines only abstract methods.

2/3 ☒ C A class can implement multiple interfaces.

1/3 ☒ D All interface methods are implicitly public.

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

3/3 ☒ A True

0/3 ☐ B False

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

1/3 ☐ A True

2/3 ☒ B False

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

0/3 ☐ A True

3/3 ☒ B False

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

2/3 ☒ A True

1/3 ☐ 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 anon5210a3005f834954

✗ public String eatCake(int number);

Anon anon5bdad77340c44d85

✗ public String abstract(NumberOfSlices);

Anon anoneee1617007114197

✗ public abstract String eatCake(int numberOfSlices) { String string = "You need to eat " + numberOfSlices + " of pizza tonight.";

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

0/3 ☐ A JavaRockStar

0/3 ☐ B new

3/3 ☒ C rockstar

0/3 ☐ D none of the above

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

3/3 ☒ A JavaRockStar

0/3 ☐ B new

0/3 ☐ C rockstar

0/3 ☐ D none of the above

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

- 2/3 ☐ A JavaRockStar
- 0/3 ☐ B rockstar
- 1/3 ☒ C SuperStarCoder
- 0/3 ☐ D new
- 0/3 ☐ E none of the above

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

- 2/3 ☒ A JavaRockStar
- 0/3 ☐ B rockstar
- 1/3 ☐ C SuperStarCoder
- 0/3 ☐ D new
- 0/3 ☐ E none of the above

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

- 2/3 ☐ A Line 10 and 14 will each run twice.
- 0/3 ☐ B Line 10 will run twice, line 14 will run once.
- 0/3 ☒ C This will not compile due to line 29.
- 1/3 ☐ D This will not compile due to line 27.
- 0/3 ☐ E None of the above

```
1 public abstract class Programmer {
2     public method void testProgram();
3 }
4
5 public class SuperStarCoder extends Programmer {
6     public void testProgram() {
7         System.out.println("The testing program using ALL the languages");
8     }
9 }
10 public void testProgram() {
11     System.out.println("I have looked on languages C, Java, C#, Ruby, Perl, JavaScript, PHP, Haskell, R, R#, ...");
12 }
13
14 public class ProgrammerTest {
15     public static void main(String[] args) {
16         SuperStarCoder superStarCoder = new SuperStarCoder();
17         superStarCoder.testProgram();
18         superStarCoder.testProgram();
19         Programmer programmer = new Programmer();
20         programmer.testProgram();
21         programmer.testProgram();
22     }
23 }
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

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

- 0/3 ☐ A public abstract interface class Payable {}
- 0/3 ☐ B public abstract Payable {}
- 0/3 ☐ C public abstract interface class Payable extends Payable {}
- 3/3 ☒ D public interface Payable {}
- 0/3 ☐ E interface PayMe() extends Money implements Payable()

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

- 1/3 ☐ A True
- 2/3 ☒ B False

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

3/3 ☒ A True

0/3 ☐ B False

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

3/3 ☒ A True

0/3 ☐ B False