socrative

Java 111 Chapter 8 (from version 1)

17 Questions

1. An abstract class can only have abstract methods.

2/4 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.

3/4 A True

1/4 B False

3. Which of the following are true?

0/4 A An interface must be created using the keyword "abstract".

3/4 B An interface defines only abstract methods.

3/4 C A class can implement multiple interfaces.

4/4 D All interface methods are implicitly public.

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

4/4 A True

0/4 B False

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

1/4 A True

3/4 B False

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

1/4 A True

3/4 B False

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

4/4 **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 anonb4f621c61f554407** / **Anon anonb19e03215b64458b**

2/4 public abstract String eatCake(int numberOfSlices);

✗ **Anon anona8e331bcfc004d43**

1/4 public abstract String eatCake(int slices)(){};

✗ **Anon anon8b5475f68d344885**

1/4 public abstract void eatCake(String numSlices);

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

1/4 **A** JavaRockStar

0/4 **B** new

3/4 **C** rockstar

0/4 **D** none of the above

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

4/4 **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?

2/4 **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?

4/4 **A** JavaRockStar

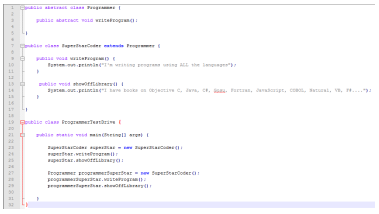
0/4 **B** rockstar

0/4 **C** SuperStarCoder

0/4 **D** new

0/4 **E** none of the above

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



2/4 **A** Line 10 and 14 will each run twice.

0/4 **B** Line 10 will run twice, line 14 will run once.

2/4 **C** This will not compile due to line 29.

0/4 **D** This will not compile due to line 27.

0/4 **E** None of the above

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

0/4 **A** public abstract interface class Payable {}

0/4 **B** public abstract Payable {}

0/4 **C** public abstract interface class Payable extends Payable {}

4/4 **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

4/4 **B** False

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

4/4 **A** True

0/4 **B** False

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

4/4 **A** True

0/4 **B** False