socrative by MasteryConnect

Java 111 Chapter 8 (from version 1)

Total Questions: 17

Most Correct Answers: **#16**

Least Correct Answers: **#8**

1. An abstract class can only have abstract methods.

2/12 (A) True

9/12 (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.

10/12 (A) True

1/12 (B) False

3. Which of the following are true?

4/12 (A) An interface must be created using the keyword "abstract".

7/12 (B) An interface defines only abstract methods.

8/12 (C) A class can implement multiple interfaces.

6/12 (D) All interface methods are implicitly public.

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

7/12 (A) True

4/12 (B) False

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

3/12 (A) True

8/12 (B) False

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

2/12 (A) True

9/12 (B) False

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

11/12 (A) True

0/12 (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 anon0b376cd0389d4bd9

✗ public abstract eatCake(int numberSlices);

Anon anon63e26299e1884b42

✗ public abstract String eatCake(numberOfSlices);

Anon anon707f6632522c44cb

```
public String abstract eatCate(int numSlices)
{
```

✗ }

Anon anon70d5ee89101f4ba7

✗ public abstract eatCake(int numberOfSlices);

Anon anonad2c8d8d1f974697

```
public abstract String eatCake(String numberOfSlices) {
return "YUM"
```

✗ }

Anon anonba7816ceadff4125

✗ public abstract String eatCake(int numberSlice);

Anon anonbb56d6c9249b4426

✗ public abstract String eatCake(int SlicesToEat);

Anon anonc1b092d992954bb1

```
public abstract eatCake(int numberSlices) {
return "Eat cake";
```

✗ }

Anon anonc34814ce53cc40d0

```
public abstract String eatCake(String numberOfSlices) {
return numberOfSlices;
```

✗ }

Anon anonde6a3cc1d3dc4960

```
public abstract String eatCake(Int numberOfSlices) {
return "Eating cake";
```

✗ }

Anon anonf6cb6fb679ca4183

✗ public abstract String eatCake(int slices);

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

0/12 A JavaRockStar

0/12 B new

11/12 C rockstar

0/12 D none of the above

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

11/12 A JavaRockStar

0/12 B new

0/12 C rockstar

0/12 D none of the above

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

0/12 A JavaRockStar

0/12 B rockstar

11/12 C SuperStarCoder

0/12 D new

0/12 E none of the above

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

10/12 A JavaRockStar

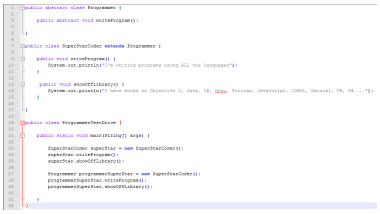
0/12 B rockstar

1/12 C SuperStarCoder

0/12 D new

0/12 E none of the above

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



4/12 A Line 10 and 14 will each run twice.

2/12 B Line 10 will run twice, line 14 will run once.

4/12 C This will not compile due to line 29.

2/12 D This will not compile due to line 27.

0/12 E None of the above

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

0/12 A public abstract interface class Payable {}

2/12 B public abstract Payable {}

0/12 C public abstract interface class Payable extends Payable {}

10/12 D public interface Payable {}

0/12 E interface PayMe() extends Money implements Payable()

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

2/12 A True

10/12 B False

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

12/12 A True

0/12 B False

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

9/12 A True

3/12 B False