socrative

Java 111 Chapter 7 (from version 1)

13 Questions

1. A subclass inherits all the fields and methods from its superclass.

2/3 **A** True

1/3 **B** False

2. Which of the following are true?

3/3 **A** You can write a new instance method in the subclass that has the same signature as the one in the superclass, thus overriding it.

3/3 **B** You can declare new methods in the subclass that are not in the superclass.

3/3 **C** You can declare new fields in the subclass that are not in the superclass.

0/3 **D** None of the above.

3. The 'IS-A' test can be used to determine whether an item:

3/3 **A** is a subclass

0/3 **B** should be a method

0/3 **C** is an attribute

0/3 **D** should be overridden

4. Considering the code, which of the following is true? public class Cat extends Animal { . . . }

0/3 **A** Cat 'has-a' Animal

3/3 **B** Cat 'is-a' Animal

0/3 **C** Animal 'is-a' Cat

0/3 **D** Cat 'has-a' Cat

5. A kumquat is:

1/3 **A** An orange

0/3 **B** A Grape

1/3 **C** Both

1/3 **D** Citrus

6. A subclass can use the "extends" keyword to extend multiple superclasses.

2/3 **A** True

1/3 **B** False

7. A superclass called Fruit contains a method called display() that outputs a message to the terminal. Apple is a subclass of Fruit. Which code segment IN THE SUBCLASS Apple will successfully call that method?

1/3 **A** super.display();

2/3 **B** Fruit.display();

0/3 **C** display(Fruit);

0/3 **D** Apple.display();

8. Which of the following are true with regard to inheritance?

0/3 **A** A subclass must have methods or instance variables in its source code

3/3 **B** A subclass is can be a superclass to another class

3/3 **C** A subclass is able to override methods of a superclass

3/3 **D** A subclass inherits instance variables and methods from a superclass

9. A superclass with a method that has the same header as a subclass will override the subclass' method.

0/3 **A** True

3/3 **B** False

10. You have two classes shown below. If you run a test drive and create a Surgeon object and call the treatPatient method, what will the output be?

```
public class Doctor {

    boolean worksAtHospital;

    void treatPatient() {
        System.out.println("Pop some pills");
    }
}

public class Surgeon extends Doctor {

    void treatPatient() {
        System.out.println("I'm going to put you under and perform surgery");
    }

    void makeIncision() {
        System.out.println("Snip, Snip");
    }
}
```

0/3 **A** Pop some pills

0/3 **B** Pop some pills I'm going to put you under and perform surgery

3/3 **C** I'm going to put you under and perform surgery

0/3 **D** I'm going to put you under and perform surgery Snip, Snip

11. Method "overloading" is when:

0/3 **A** Two or more methods have the same name, and the same number and type of parameters, but different return types

3/3 **B** Two or more methods have the same name but different numbers or types of parameters

0/3 **C** A method can take any number of arguments of the same type

0/3 **D** A method calls the method of the same name of its superclass

12. Assuming the upper code is part of a Thing class and the lower code has an instance of the class called thing, what will the result be?

```
...
public String getValue() {
    return "20";
}
public int getValue() {
    return 20;
}
...
System.out.println("Twice thing's value is: "
        (thing.getValue() + thing.getValue()) );
...
```

0/3 **A** Twice thing's value is: 40

2/3 **B** Twice thing's value is: 2020

0/3 **C** The code will not run

1/3 **D** The code will not compile

13. Write the method header for a method named "playRecord" that accepts three parameters: an int for the track number, an int for the speed (rpms), and a String for the track title. The method will not return anything.

✗ **Anon anondaf1ce42d8bf4011**

1/3

```
public void playRecord(int trackNumber, int speedRPM, String trackTitle)
```

✗ **Anon anone129b452a63741b4**

1/3

```
void playRecord(int, rpms, String) {}
```

✗ **Anon anonf1ac7f49a13a48b1**

1/3

```
public void playRecord(int trackNumber, int speed, String trackTitle) {
}
```