

Pencil & Paper 14

1. Is the class Media below a base/superclass, a derived/subclass, or both? If it is a base/superclass, what classes are its subclasses (your answer should state "Media is a superclass to _____" or "Media has no subclasses")? If it is a derived/subclass, what class is its superclass ("Media is a subclass of _____")?

The class Media is a base/superclass because the keyword "extends" is not a part of its class declaration. Media is a superclass to the Book, Movie, and Game classes because the keyword "extends" is followed by the superclass Media in their class declarations. (Note: the class VideoGame is a derived/subclass of Game, which itself is a derived/subclass of Media).

2. Is the class Book below a base/superclass, a derived/subclass, or both? If it is a base/superclass, what classes are its subclasses (your answer should state "Book is a superclass to _____" or "Book has no subclasses")? If it is a derived/subclass, what class is its superclass ("Book is a subclass of _____")?

The class Book is a derived class/subclass. It is a subclass of Media.

3. Is the class Game below a base/superclass, a derived/subclass, or both? If it is a base/superclass, what classes are its subclasses (your answer should state "Game is a superclass to _____" or "Game has no subclasses")? If it is a derived/subclass, what class is its superclass ("Game is a subclass of _____")?

The class Game is both a derived class/subclass and a base class/superclass. It is a subclass of Media while it is a superclass to VideoGame.

4. Is the class VideoGame below a base/superclass, a derived/subclass, or both? If it is a base/superclass, what classes are its subclasses (your answer should state "VideoGame is a superclass to _____" or "VideoGame has no subclasses")? If it is a derived/subclass, what class is its superclass ("VideoGame is a subclass of _____")?

The class VideoGame is a derived class/subclass. It is a subclass of Game (which itself is a subclass of Media).

5. List all of the public instance variables and public methods of the class Book, including all of those it inherits from its superclass.

The class Book inherits the public methods setName, getName and toString, while it declares the new methods setAuthor and getAuthor. Book does not have any instance variables that are specifically declared as public, but it does inherit the protected variable status from its superclass Media. Because Book is a subclass of Media and status is declared in the superclass as protected, the variable can be accessed in Book and behaves the same as if it were declared as public.

6. List all of the public instance variables and public methods of the class VideoGame, including all of those it inherits from its superclass.

The class VideoGame inherits the public methods setMaxNumberOfPlayers, getMaxNumberOfPlayers, setName and getName, while it declares the new methods setUsed and getUsed (it also overrides the toString method). As in question 5, VideoGame does not have any instance variables that are specifically declared as public, but it does

inherit the protected variable status from its superclass Game. Because VideoGame is a subclass of Game (which itself is a subclass of Media), and because status was originally declared as protected, the variable can be accessed in VideoGame and behaves the same as if it were declared as public.

7. Which of the following two relationships between classes is appropriate for inheritance?

Box has-a Surprise

Box is-a Container

The relationship between a box and a container is more appropriate for inheritance since a box is a particular form of container much more than it is a form of surprise. If one were to write a class for a new type called Container, it would be natural to create a new subclass of Container called Box (perhaps along with Jar, Barrel, Can, etc.). By structuring the code in this way an entire set of related classes could be made that share the common variables and methods that were defined in the superclass Container.

8. List the child classes of the class Media.

The child classes (or descendent classes) of Media are Book, Movie, Game, and VideoGame.

9. What does it mean for a child class to override a method that it inherits from a parent class?

According to our textbook, a method is overridden when “the definition of an inherited method [is] changed in the definition of the derived class so that it has a meaning in the derived class that is different from what it is in the base class” (439). In other words, the derived class replaces the definition of a method that it inherits from its superclass.

10. Is overriding a method the same as overloading a method?

No, overriding replaces an inherited method definition from a superclass while method overloading allows for many methods with different parameters and definitions to share the same name.

11. Does the class Book below override the toString method?

The class book does not override the toString method; it uses the method as it was defined in its superclass Media.

12. What method(s) does the class VideoGame below override?

The class VideoGame overrides the toString definition inherited from its superclass Game.

13. Can a child class directly access a private member (method or instance variable) of a parent class?

No, it cannot. According to our textbook, private variables are not inherited and “an instance variable (or method) that is private in a base class is not directly accessible outside of the definition of the base class, not even in a method definition for a derived class. The private methods of the base class are just like private variables in terms of not being directly available. But in the case of methods, the restriction is more dramatic. A private variable can be accessed indirectly via an accessor or mutator method. A private method is simply not available. It is just as if the private method were not inherited” (454-455). Private variable can only be accessed indirectly via inherited methods but private methods in the superclass cannot be accessed at all.

14. What is the UML notation to show that one class is derived from another class?

According to the article “Practical UML: A Hands-On Introduction for Developers”, the UML

notation to communicate inheritance is called a “generalization” and is represented by a triangle pointing to the superclass from the subclass.

15. Look in the overloaded constructor for the class Book below. What does super refer to in this method?

In the class Book “super(newName)” is called within its overloaded constructor, which invokes the overloaded constructor for the superclass Media. Because the variable name is private and not inherited by Book, this is an indirect means of changing its value to newName.

16. Look in the overridden toString method in the class Movie below. What does super.toString() refer to in this method?

In the class Movie “super.toString()” invokes the toString() method of its superclass Media. The toString() method was overridden by Movie and this serves as a means to invoke the original toString() method from the superclass.

17. Is the following code allowed in Java, assuming it is written inside a complete and correct driver?

```
Media m = new Book("Moby Dick", "Herman Melville");
```

Yes, this is valid. Because Book is a subclass of Media, objects of Book can be assigned to variables of type Media (but not vice versa). According to our textbook, “An object of a derived class has the type of the derived class, and it also has the type of the base class. More generally, a derived class has the type of every one of its ancestor classes. So, you can assign an object of a derived class to a variable of any ancestor type (but not the other way around). You can plug in a derived class object for a parameter of any of its ancestor types. More generally, you can use a derived class object anywhere you can use an object of any of its ancestor types” (446).

18. What is the output of the following code, assuming it is written inside a complete and correct driver?

```
Media m = new Movie("Jaws", 124.0f);  
System.out.println(m);
```

The output is as follows:

Name: Jaws

Running time: 124.0 minutes

19. Is the following code allowed in Java, assuming it is written inside a complete and correct driver?

```
Book b = new Media("War and Peace");
```

No, this is not allowed. The variable b is of type Book, which is a subclass of Media. Instances of a superclass cannot be assigned to types of a subclass. However, it is valid to assign instances of a subclass to variable types of a superclass.

20. What is the purpose of the operator instanceof?

According to our textbook, the instanceof operator “checks to see if an object is of the type given as its second argument. The syntax is (Object instanceof Class_Name) which returns true if Object is of type Class_name; otherwise it returns false” (467). This can be used in many ways, such as to verify that two objects assigned to two different variables of a shared superclass are also of the same type themselves.

21. Will the following code compile, assuming it is written inside a complete and correct driver? Briefly explain your answer.

```
Media m = new Game("Monopoly", 4);  
m.setMaxNumberOfPlayers(6);
```

No, it will not compile. The method `setMaxNumberOfPlayers` is defined only in the class `Game`, which is a subclass of `Media`. In order for this line of code to compile correctly the variable `m` must be of type `Game`. Another potential solution is to define the method `setMaxNumberOfPlayers` in the class `Media` (although it would allow the code to compile, doing this would not make very much sense).

22. Will the following code compile, assuming it is written inside a complete and correct driver?

```
Media m = new Game("Monopoly", 4);  
if (m instanceof Game)  
{  
    Game tempReferenceToGame = (Game)m;  
    tempReferenceToGame.setMaxNumberOfPlayers(6);  
}
```

Yes, this would compile. The instance of `Game` is first assigned to a variable of type `Media`, then it is cast to a different variable of type `Game`. Variables of type `Game` can "see" the additional method definitions in the `Game` subclass and therefore can invoke the method `setMaxNumberOfPlayers`.

23. What does it mean that the member variable status in the class `Media` is protected?

Because `status` is declared as "protected" it becomes visible to all derived classes from the superclass as if it were declared as `public`. According to our textbook, "If a method or instance variable is modified by `protected` (rather than `public` or `private`), then it can be accessed by name inside its own class definition, by name inside any class derived from it, and by name in the definition of any class in the same package" (456).

24. Is the following code allowed in Java, assuming it is written inside a complete and correct driver?

```
Book b1 = new Book();  
System.out.println(b1.status);
```

Yes, this is allowed. The variable `status` is inherited by the class `Book` as if it were a variable that was declared as `public` and so it can be accessed by any part of the program that can access the variable `b1`.

25. Look in the overridden `toString` method in the class `VideoGame` below. Is the direct reference to `status` allowed here?

Yes, this is allowed. The parent class of `VideoGame` is `Game`, which itself is a subclass of `Media`. Because the variable `status` is declared as `public` in `Media`, it is inherited as a `public` variable by all the child classes of `Media`. Therefore, it is accessible anywhere inside of the `VideoGame` class.

26. What is the EXACT output of this week's "Coding" files after they have been compiled and executed (NOTE: there are some random numbers in the output so report EXACTLY what you see when you compile and run)?

The following output was printed to the console:

```
++++  
Name: Play-doh [1]  
++++  
Name: The Hobbit [3]
```

Twas a dark and stormy night...
++++
Name: Star Wars [5]
Running time: 121.0 minutes
Prepare yourself for 121.0 minutes of action, adventure, mystery, or romance!
++++
Name: Battleship [7]
Max players: 2
Rolling the dice, I get a 5 and a 2
++++
Name: Pac Man [9]
Max players: 1
Used
Must be a classic
++++
Name: NO NAME [0]
++++
Name: NO NAME [2]
Twas a dark and stormy night...
++++
Name: NO NAME [4]
Running time: 0.0 minutes
Prepare yourself for 0.0 minutes of action, adventure, mystery, or romance!
++++
Name: NO NAME [6]
Max players: 1
Rolling the dice, I get a 6 and a 3
++++
Name: NO NAME [8]
Max players: 1
New
Hope it is a classic