Chapter 9: Polymorphism

Multiple Choice Questions:

1) A polymorphic reference is one that can refer to _____ type(s) of object(s).

a) exactly one
b) zero
c) multiple
d) abstract
e) static

2) The commitment to execute certain code to carry out a method invocation is referred to as _____.

a) execution
b) binding
c) polymorphism
d) inheritance
e) none of the above

3) In Java, polymorphic method binding occurs _____ .

a) at run time
b) at compile time
c) never
d) when a programmer writes the code
e) during the testing phase of software development

4) Late binding is _____ than _____ .

a) more efficient, compile-time binding
b) less efficient, compile-time binding
c) more efficient, run-time binding
d) less efficient, run-time binding
e)

5) Suppose that `Horse` is a subclass of `Animal`, and neither class is abstract. Which of the following is an invalid declaration and initialization?

a) `Horse h = new Horse();`
b) `Horse h = new Animal();`
c) `Animal a = new Animal();`
d) `Animal a = new Horse();`
e) all of the above are valid

6) In Java, a(n) _____ is a collection of constants and abstract methods.

a) polymorphic reference
b) abstract class
c) implementation
d) interface
e) iterator

7) In Java, polymorphic references can be created through the use of _____ and _____.

a) inheritance, interfaces
b) inheritance, abstract classes
c) interfaces, abstract classes
d) interfaces, iterators
e) none of the above

8) Let Dog be a subclass of Animal, and suppose Animal has a method called `speak()` that is overridden in the Dog class. Consider the following code.

```
Animal spot = new Dog();
spot.speak();
```

Which of the following is true?

a) This code will result in a compile-time error.
b) This code will result in a run-time error.
c) The `speak` method defined in the `Animal` class will be called.
d) The `speak` method defined in the `Dog` class will be called.
e) The `speak` method will not be called at all.

9) The `Comparable` interface contains which of the following methods?

a) `isGreaterThan`
b) `isLessThan`
c) `equals`
d) `compareTo`
e) all of the above

10) Let `Object a` be larger than `Object b`. What will the following method call return?

```
a.compareTo(b)
```

a) it will return 0
b) it will return a number greater than 0
c) it will return a number less than 0
d) it will return true
e) it will return false

11) Which of the following methods are included with any object that implements the `Iterator` interface?

a) `next`
b) `hasNext`
c) `toString`
d) all of the above
e) a and b

12) You need to create a reference variable that can refer to objects from many different classes. You do not know the inheritance hierarchies of the classes. The safest class to use to declare the reference variable is

a) `Animal`
b) `String`
c) `Object`
d) `Scanner`
e) `File`

13) Consider the following line of code.

```
Comparable s = new String();
```

Which of the following statements is true about this line?

a) It will result in a compile-time error.
b) It will result in a run-time error.
c) It will create a `String` object pointed to by a `Comparable` reference.
d) Although it is perfectly valid Java, it should be avoided due to confusion.
e) none of the above are true

14) Suppose `Animal` is an interface that specifies a single method – `speak`. Now suppose the Dog class implements the `Animal` interface. In addition to the speak method, the Dog class also has a method called `wagTail`. Now consider the following code.

```
Animal a = new Dog();
a.wagTail();
```

Which of the following is true about this code?

a) It will result in a compile-time error.
b) It will result in a run-time error.
c) It will call the `speak` method defined in the Animal interface.
d) It will call the `wag`Tail method defined in the Dog class.
e) none of the above are true.

15) Which GUI concepts use polymorphism to establish their relationship?

a) a listener and its associated component
b) a radio button and its default selection
c) a button and its label
d) a slider and its tick marks
e) none of the above

True/False Questions:

1) Consider a reference declared in the following manner.

```
Animal a;
```

This reference may only point to an object that created by instantiating the `Animal` class.

2) Let `Animal` be an interface. Then it is possible to create an object by instantiating the `Animal` interface.

3) The `compareTo` method of the `Comparable` interface returns a boolean value.

4) Compile-time binding is more efficient than dynamic binding

5) A parameter to a method can be polymorphic.

6) An interface cannot declare any instance variables.

7) Establishing the relationship between a listener and the component it listens to is accomplished using polymorphism.

8) A reference variable can refer to an object of a child class, but not any further down the inheritance hierarchy.

9) Polymorphism via inheritance requires that all classes in the inheritance hierarchy are concrete.

10) An interface name may be used as a reference type.

Short Answer Questions:

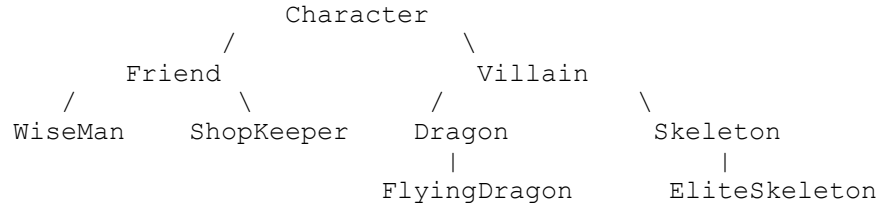
1) What is polymorphism?

2) How does inheritance relate to polymorphism in Java?

3) Consider a class hierarchy that includes a class called `Vehicle`, with subclasses called `Car` and `Airplane`. The `Vehicle` class has a method called `getMaxSpeed`, which is overridden in the `Car` class. The `getMaxSpeed` of the `Vehicle` class returns 760 mph, while the `getMaxSpeed` method of the `Car` class is overridden to return 150 mph. What is the output of the following snippet of code? Explain your answer.

```
Vehicle v = new Car();
System.out.println(v.getMaxSpeed() + " mph");
```

4) Consider the following inheritance hierarchy that is used in a video game.



Which of the following declarations and initializations will not cause a compiler error?

```
Character c = new FlyingDragon();
FlyingDragon f = new Character();
Dragon d = new Villain();
Villain v = new Skeleton();
Dragon d = new ShopKeeper();
```

5) When a reference variable refers to an object that is in an inheritance hierarchy and a method of the object is invoked, how does Java determine which version of the method to use?

6) Are there any differences between extending a class and implementing an interface?

7) Describe the `compareTo` method and the circumstances under which it returns different values.

8) Does polymorphism work if some of the classes in an inheritance hierarchy are abstract?

9) Can an interface hierarchy be used for polymorphism? Explain.

10) Write an interface for a CD player. It should have the standard operations (i.e. play, stop, etc) that usual CD players have.

11) Give an example of a class that implements the `Comparable` interface, and explain how the implementation of the `compareTo` method determines its return value.

12) Is dynamic binding used when polymorphic references are made using interfaces?

13) Can a polymorphic reference invoke a method that is only declared at the object's class level? If "yes", explain how.

14) Suppose you are implementing the comparable interface in a class representing a `Person`, where the ordering is based on the age of the person. Write a `compareTo` method for this class. You may assume that there is an instance variable called `age` and an accessor method called `getAge`.

15) Why can't an interface be instantiated?