CS1410
Polymorphism Lab

For each term in the left column, write the letter for the description from the right column that best matches the term.

Fill in the blanks for each of the following statements:

1 _I__ **`abstract`** method
2 _J__ **`getClass`** method
3 _H__ **`implements`** keyword
4 _L__ type-wrapper classes
5 _E__ downcasting
6 _K__ concrete class
7 _F__ polymorphism
8 _D__ **`instanceof`**
9 _B__ **`final`**
10 _C__ **`getName`** method
11 _G__ **`abstract`** class
12 _A__ interface

a) Can be used in place of an **`abstract`** class when there is no default implementation to inherit.
b) Indicates that a method cannot be overridden or that a class cannot be a superclass.
c) **`Class`** method which returns the name of the class associate with the **`Class`** object.
d) An operator that returns true if its left operand (a variable of a reference type) has the *is-a* relationship with its right operand (a class of interface name).
e) Uses superclass references to manipulate sets of subclass objects in a generic manner.
f) Casting a superclass reference to a subclass reference.
g) Cannot be instantiated; used primarily for inheritance.
h) Indicates that a class will declare each method in an interface with the signature specified in the interface declaration.
i) Must be overridden in a subclass; otherwise, the subclass must be declared **`abstract.`**
j) Returns an object that can be used to determine information about the object's class.
k) A class that can be used to create objects.
l) Classes in the **`java.lang`** package that are used to create object containing values of primitive types.

13. With _Polymorphism__, it becomes possible to design and implement systems that are more extensible.

14. Although we cannot instantiate objects of `abstract` superclasses, we can declare _Variables__ of abstract superclass types.

15. It is a syntax error if a class with one or more `abstract` methods is not explicitly declared _abstract___.

16. It is possible to assign a superclass reference to a subclass variable by _defining__ the reference to the subclass type.

17. A(n) _Abstract Class_ may contain a set of `public abstract` methods and/or `public static final` fields.

18. When a method is invoked through a superclass reference to a subclass object, Java executes the version of the method found in the _subclass__.

19. The _instanceof__ operator determines whether the type of the object to which its left operand refers has an *isa* relationship with the type specified as its right operand.

20. To use an interface, a class must specify that it _implements_ the interface and must declare every method in the interface with the signatures specified in the interface declaration.

21. When a class implements an interface, it establishes an _is-a__ relationship with the interface type.

In the space provided, answer each of the given questions. Your answers should be concise; aim for two or three sentences.

22. Describe the concept of polymorphism.

Polymorphism allows programming in the general, meaning that it allows processing of objects that all share the same super class.

23. Define what it means to declare a method `final` and what it means to declare a class `final`.

Final Methods cannot be overridden in a subclass, the implementation of the method can never change.

Final Classes cannot be a superclass, all methods and items in the class are implicitly final.

24. What happens when a class specifies that it implements an interface, but does not provide declarations of all the methods in the interface?

The compiler would return a compilation error stating that the class must be declared abstract.

25. Describe how to determine the class name of an object's class.

The getName method can be used to return the class name of the object's class.

26. Distinguish between an abstract class and a concrete class.

An abstract class is meant to be very general for classes that share attributes, thus it cannot be used to instantiate an object.

A concrete class is meant to be very specific, it inherits from the abstract class in order to share attributes with other objects, which it can use to instantiate an object.