Which statements are true about Class Diagrams?

A. From the conceptual perspective, associations represent relationships between classes.

B. UML can be used only to model Object-oriented systems.

C. Multiplicities are normally 0, 1, \*. It does not support for a range like 2-4.

D. Within the specification perspective, associations represent methods.

F. From specification perspective, association lines with arrows indicate navigability. The source

class has responsibility of 'knowing' the target class but not the other way round.

A-F

The fact that the same operation may apply to two or more classes is called -----------?

Choose one

A. Inheritance

B. Polymorphism

C. Encapsulation

D. Multiple classification

Polymorphism

Which of the following about an abstract class is correct?

A. A class that has direct instances, but its descendants may have direct instances.

B. A class that has no direct instances, but its descendants may have direct instances

C. A class that has direct instances, but its descendants may not have direct instances.

D. A class that has no direct instances, but its descendants may not have direct instances

B. because, if its descendants are also abstract, they also cannot have direct instances, because we don’t have an abstract class that all its descendants are also abstract

14. In the following code, which of the following is correct regarding the relationship between Employer and Gardener?

A. There is a dependency from Employer to Gardener

B. There is a one-way association from Employer to Gardener

C. There is a two-way association between Employer and Gardener

D. Not possible to determine from the code shown.

public class Employer {

public void employ() {

Gardener gardene= new Gardener() ;

gardener.garden();

}

}

public class Gardener {

public void garden() {

// do gardening

}

}

A. it's dependency, we don’t have a reference of Gardener in the Employee class, we create an object from Gardener inside a method to call a method from the Gardener  class

Explain the concept of polymorphism.

Polymorphism is one of the [OOPs](https://beginnersbook.com/2013/04/oops-concepts/) features that allow us to perform a single action in different ways.

Explain how to turn a class into an immutable class.

1. Declare the class as final so it can’t be extended.
2. Make all fields private so that direct access is not allowed.
3. Don’t provide setter methods for variables.
4. Make all mutable fields final so their values can be assigned only once.
5. Initialize all the fields via a [constructor](https://www.journaldev.com/18899/constructor-in-java) performing the deep copy.
6. Perform [cloning](https://www.journaldev.com/60/java-clone-object-cloning-java) of objects in the getter methods to return a copy rather than returning the actual object reference.

What is the difference between Abstract and Final class and method :

abstract class: no object

abstract method: no body

final class: no child

final method: no override