

Question-Answer Session

Q: What is the difference between the static type of a variable and the dynamic type of a variable?

A: The static type of a variable *v* is the type as declared in the source code in the variable declaration statement. The dynamic type of a variable *v* is the type of the object that is currently stored in *v*. For example, if `Vehicle v1=newCar();` then the static type of *v1* is `Vehicle`, and the dynamic type of *v1* is `Car`.

Q: What is overriding?

A: A subclass can override a method implementation. In order to do this, the subclass declares a method with the same signature as the superclass, but with a different method body. The overriding method takes precedence when a method is called on subclass objects.

Q: What is method polymorphism (or polymorphic method dispatch)?

A: Method calls in Java are said to be polymorphic, which means that the same method call can at different times invoke different methods. This depends on the dynamic type of the variable that we use to make that call. For example, in the DoME example program, the statement `item.print();` could invoke the `CD`'s `print` method at one time and the `video`'s `print` method at another.

Q: What is the `toString` method?

A: Each object in Java has a `toString` method. When it is called, it can be used to return a `String` representation of the method. In order for it to be useful, an object should override this method.

Q: What happens when we declare a method or field protected?

A: This will allow direct access to this field or method from direct or indirect subclasses.

Q: So what is the difference between declaring a method protected, public, or private?

A: With a public method, any other class can access it. A private method can only be accessed within its own class. A protected method can be called within a class itself and from all its subclasses, but not to other classes.

Q: What's all this obsession with Frank Sinatra?

A: I honestly don't know.