Interfaces

In addition to the class, Java has a construct called the *interface*. Interface C1 is shown to the right. Instead of keyword **class**, use **interface**. The components of the class *must* be **public**, **abstract** methods, as shown (we mention two other possibilities in a later video). So an interface looks a lot like an abstract class all of whose components are abstract methods. You can think of it that way.

```
public interface C1 {
   public abstract int m() ;
   public abstract int p();
}
```

Since the methods *must* be **public** and **abstract**, those keywords can be omitted, as you see in the declaration of interface C2.

```
public interface C2 {
   int m() ;
   int q();
}
```

Having written interfaces C1 and C2, we can write a class A that implements them, using keyword **implements** instead of **extends**. Here are two rules:

1. If a class implements an interface, it must declare (override) all methods that are declared in the interface.
2. A class can implement more than one interface.

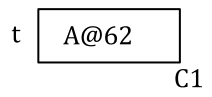
```
public class A implements C1, C2 {
   public int m() {...}
   public int p() {...}
   public int q() {...}
   public void t() {
      int x= m();
         ...
   }
   ...
}
```

Class A does declare all the methods of the two interfaces.

Method m(() appears in both interfaces. There is no ambiguity because no implementation of the methods is given in the interfaces, only the syntax.

We cannot create an instance of an interface. But we can assign a class A object to a C1 variable —C1 can be used as a type:

C1 t= **new** A(…);



Thus, t can contain a pointer to an A object. From t’s perspective, the only methods that can be called are those declared in interface C1 or superclass Object. More on this later.

An interface can extend another interface —but use keywords **extends**, not **implements**:

public interface C3 **extends** C1 { … }.

A class that implements C3 must declare all methods in both C3 and C1.

That’s the definition of an interface

We’ve shown you what an interface is and how one uses it. We haven’t yet given you really good examples of its use. That will come! Be patient.