Interfaces—Quick Class

1. Interfaces are something like abstract classes except they do not have constructors
2. Classes implement interfaces. The can implement more than one interface.

   --Assume that A and B are interfaces:

   class C implements A,B{

   }
3. Interfaces extend interfaces

   Interface C extends A,B{

   }
4. Fields: All fields in an interface are intrinsically public static and final.
   - For example int n = 12; would compile but it is public static final

There are no instances variables in an interface

5. Abstract methods (no {}) All abstract methods are implicitly public abstract.
   a. For example: int add(int n): is public abstract
6. Default methods. These are essentially instance methods. Example:
   a. (public) default int add(int n){

      return n+2;

      }

   Note: implicitly public, they can be overwritten in a class or sub interface

b. There now can also be private default methods ex:

Example: private default int sub(int n){

return n-2; }

7. Static Methods:
   a. (public) static int add(int n){
      return n+n;} // would be called A.add(2);
   b. private static int add(int n){
      return n+n;}