Winston Wang

Vocab 1/23/15

1. Public Interface of a class

**public** **class** Foo

{

**public** **int** hello;

**private** **double** bye;

**public** **void** hi()

{

}

**public** **double** getBye()

{

**return** bye;

}

}

The public interface of a class is the part of the class that is public. This includes all methods and public fields. Private fields can be accessed by the public interface through getters and setters

1. Interface

**interface** Foo

{

**void** hi();

**void** bye();

**int** count();

}

If a class implements an interface, then it must implement all of the methods that appear in that interface

1. Encapsulation

**public** **class** Foo

{

**private** **int** hello;

**private** **double** bye;

**public** **int** getHello()

{

**return** hello;

}

**public** **double** getBye()

{

**return** bye;

}

**public** **void** setBye(**double** b)

{

bye = b;

}

}

The code above shows encapsulation in the Foo class. Both bye and hello can be accessed using getters, but hello cannot be manipulated from the outside, since it lacks a setter

1. Invariant

**public** **class** Foo

{

**public** **static** **void** main(String args[])

{

**int** invar = 50;

**int** m = 100;

**for** (**int** i = 0; i < 100; i++)

{

invar = (m + i)/2;

m--;

System.***out***.println(invar);

}

}

}

Although the example above is a loop invariant, invariants are not limited to loops. Certain expressions, such as , are always 0 (from physics).