Jenny Hamer

CIS 36B

Lab 5 : Interfaces

Answer the following questions at end of chapter 8:

1, 2, 3, 4, 5, 7, 8, 9, 10

1. “One interface, multiple methods” is a key tenet of Java. What feature best exemplifies it?

Polymorphism best exemplifies this tenet.

2. How many classes can implement an interface?

An interface can be implemented by any number of classes.

3. How many interfaces can a class implement?

Technically, there isn’t a limit on the number of interfaces a class can implement.

4. A class declares that it implements an interface by use of a/an: **implements clause**

5. Can interfaces be extended?

Yes, in Java an interface can extend another interface.

7. Variables declared in an interface are implicitly **static** and **final**. What good are they?

Variables declared in an interface essentially serve as constants and must be initialized. In large programs, it’s useful to use them to describe constants like array sizes, different limits (ie. min(), max()), and other constant values.

8. Can one interface be a member of another?

An interface can be a member of another interface or of another class. This type of interface is known as a member interface or a nested interface.

9. Given two interfaces called **Alpha** and **Beta** show how a class called **MyClass** specifies that it implements each.

class MyClass implements Alpha, Beta { //

}

10. Create a new class **Constants** that implements the **Series** interface discussed at the beginning of this chapter. Its **getNext()** method repeatedly returns the last value passed in as an argument to the **setStart()** method. Until **setStart()** is called, it returns 0.

class Constants implements Series {

int value;

public Constant() { // constructor

value = 0;

public void setStart(int x) {

value = x;

public int getNext() {

return value;

public void reset(); {

value = 0;