Group 19

Xingxuan Zhang, Casey Cadenhead

7.1

**public** **static** **float** average (**int** num1, **int** num2)

{

**return** (num1+num2)/2;

}

7.2

**public** **static** **float** average (**int** num1, **int** num2)

{

**return** (num1+num2)/2;

}

**public** **static** **float** average (**int** num1, **int** num2,**int** num3)

{

**return** (num1+num2+num3)/3;

}

7.10

Parameters in a Java method are passed by value. A copy of the actual parameter (the value passed in) is stored into the formal parameter (in the method header).

When an object is passed to a method, the actual parameter and the formal parameter become aliases of each other.

No, it is not consistent because the method may or may not have permanent effect on the object and it just passes the reference to the method.

7.11

A static method cannot be used to reference an instance variable because with a static method, it is invoked through a class name instead of through an object and with an instance variable, each instance of the class has its own version of the variable (meaning that each object can have its own distinct value for that variable). Without invoking through the object, the method may end up referencing the wrong variable, or not knowing which version to run with.

7.12

Yes, a class can implement two interfaces each of which contains the same method signature. The class which implements an interface provides method implementations for each of the abstract methods defined in the interface. In satisfying the requirements for a method of one interface, it simultaneously satisfies the requirements for a method with the same signature in another interface.