1. When an object is passed as a parameter, it is not copied. Therefore, if you change the passed object, you change the original object. This can cause large errors to arise in programming so it is key to be cautious when passing objects.
2. When working in real life, a client will not need to know how a server functions down to the circuit board, they will only need to know how to use the given interface.
3. Mutators are messages that change an object’s state. Example 1: s1.setName(“Bill”); Example 2: s1.setScore(1,84) Accessors are messages that access an object’s state. Example 1: s1.getName(); Example 2: s1.getScore(1)
4. Public: used mostly for methods Private: used mostly for instance variables
5. Constructors are used to initialize instance variables for a freshly initialized object.
6. To allow for the suggested uses of toString
7. Changing a string value within a string and then calling to that name Example: a.makeStr(“Apples”); s = b.getStr(); //both refer to the same string value
8. A primitive type is a metaphorical box that contains a value; i.e. int. A reference type is a metaphorical box that points to an object; i.e. Scanner.
9. The null value is a value that is represented as nothing in data, this can be used to ‘wipe’ a variable.
10. A null pointer exception is when a program tries to run a method with an object that is null. Example: String name = null; System.out.println(name.length());
11. When a class template does not contain a constructor, the JVM uses the default constructor.
12. The JVM uses its default constructor to initialize the instance variable.
13. It serves as a copy or temporary version of the object; can be used when passing objects as parameters.