CS165 – Object 1 Worksheet Name_____________________________________

1) How can multiple methods within a Java class read and write the same variable?

A. Allow one method to reference a local variable of the other
B. Declare a variable of the same name in both methods
C. Add the variable to the class as a class variable
D. Pass the variable as a parameter between methods
E. None of the above

2) Which of the following statements about objects and classes are correct?

A. In Java, code and data can only exist in a class.
B. Instantiation does not require memory allocation.
C. Instantiation makes a class from an object.
D. Many objects can be made from a single class.
E. Only a single object can be made from a class.

3) Which of the following statements about public versus private are correct?

A. Public variables and methods cannot be accessed outside the class in which they are defined.
B. Private variables can be accessed outside the class only by writing ”getter” or “setter” methods.
C. Private methods cannot be non-static, but public methods can be, and both can be static.
D. Private methods comprise the ‘interface’ provided to users of the class.
E. If you instantiate a class from outside the class you can access both private and public variables.

4) Which of the following statements about static and non-static are correct?

A. Static data is also called instance data, and non-static data is called class data.
B. Only one copy of instance (non-static) exists.
C. There is a separate copy of instance data for every object that is instantiated.
D. Accessing class data using the class name instead of the object name is not a good practice.
E. Accessing instance data does not require use of the class name, if done from within the same class.

5) The code below accesses class/instance variables, which line will not compile?

```
0 public class Class {
1           String s0 = “Instance Data”;
2           static String s1 = “Class Data”;
3           public static void main(String args[]) {
4                   Class instance = new Class();
5                   System.out.println(Class.s0);
6                   System.out.println(instance.s0);
7                   System.out.println(Class.s1);
8                   System.out.println(instance.s1);
9 }
```

6) What is printed?

```
public class Peer {
  static int i = 11;
  int j = 22;
  public static void main(String args[]) {
           Peer p1 = new Peer();
                   Peer p2 = new Peer();
           p1.i = 33; p1.j = 44; p2.i = 55; p2.j = 66;
           System.out.println(p1.i+“ ”+p1.j+“ ”+p2.i+“ ”+p2.j);
  }
}
```