**Name**

**Advanced Programming in Java**

**Lab Exercise 9/25/2019**

Problems 1 – 5 refer to the following code (assume that *equals* is not an explicit, method of this class):

MoonRock myRock = new MoonRock(3, “Xeon”);

MoonRock yourRock = new MoonRock(2, “Kryptonite”);

MoonRock ourRock = new MoonRock(3, “Xeon”);

MoonRock theRock;

theRock = ourRock;

1. Does *theRock.equals(ourRock)* return a *true* or *false*?

2. Does *theRock.equals(yourRock)* return a *true* or *false*?

3. Does *theRock.equals(myRock)* return a *true* or *false*?

4. Does *myRock = = ourRock* return a *true* or *false*?

5. Does *myRock.equals(yourRock)* return a *true* or *false*?

Problems 6 – 11 refer to the following code:

public class Hotdog

{

public Hotdog( )

{ **. . .** }

public String method1(int jj)

{ **. . .** }

private void method2(String b)

{ **. . .** }

public int method3( )

{ **. . .** }

public double x;

public int y;

private String z;

}

Now suppose from within a different class we instantiate a *Hotdog* object, o*scarMeyer*.

All of the code in questions 6 – 11 is assumed to be in this other class.

6. Is *int zz = oscarMeyer.method1(4);* legal? If not, why?

7. Is *oscarMeyer.method2(“Hello”);* legal? If not, why?

8. Is *int cv = oscarMeyer.method3( );* legal? If not, why?

9. Is *int cv = oscarMeyer.method3(14);* legal? If not, why?

10. Is *oscarMeyer.z = “hotdog”;* legal? If not, why?

11. Assume the following code is inside *method1*:

method2(“BarBQ”);

Is this legal? If not, why?

12. Instantiate an object called *surferDude* from the *Surfer* class using two separate lines of code. One line should declare the object and the other line should instantiate it. (Assume no parameters are sent to the constructor.)

13. Which of the following sets of code (both purport to do the same thing) is correct?

(Assume *beco* is an object having a method (*method33*) that receives a *Circle* parameter.)

a. Circle cir5 = new Circle(10);

beco.method33(cir5);

b. beco.method33( new Circle(10) ) ;

c. Both a and b

14. What is the value of *balance* after the following transactions?

//Refer to the BankAccount class you created on p 15-8 of your textbook

BankAccount acc = new BankAccount(10, “Sally”);

acc.deposit(5000);

acc.withdraw(acc.balance / 2);

15. What’s wrong with the following code?

BankAccount b;

b.deposit(1000);

16. What’s wrong with the following code?

BankAccount b new BankAccount(32.75, “Melvin”);

b = new BankAccount(1000, “Bob”);

b.deposit(“A thousand dollars”);

17. What is printed in the following?

String myString = “Yellow”;

String yourString = “Yellow”;

String hisString = new String(“Yellow”);

String ourString = myString;

System.out.println(myString = = yourString);

System.out.println(myString = = ourString);

System.out.println( myString.equals(yourString) );

System.out.println( myString.equals(ourString) );

System.out.println( myString = = hisString );

**Project… Gas Mileage**

Create a class called *Automobile* in which you pass a gas mileage (miles per gallon)

parameter to the constructor which in turn passes it to the state variable, *mpg*. The

constructor should also set the state variable *gallons* (gas in the tank) to 0. A method called *fillUp* adds gas to the tank. Another method, *takeTrip*, removes gas from the tank as the result of driving a specified number of miles. Finally, the method *reportFuel* returns how much gas is left in the car.

Test your *Automobile* class by creating a *Tester* class as follows:

public class Tester

{

public static void main( String args[] )

{

**//Create a new object called *myBmw*. Pass the constructor an**

**/ /argument of 24 miles per gallon**

Automobile myBmw = new Automobile(24);

**//Use the *myBmw* object to call the *fillup* method. Pass it an argument**

**//of 20 gallons.**

myBmw.fillUp(20);

**//Use the *myBmw* object to call the *takeTrip* method. Pass it an**

**//argument of 100 miles. Driving 100 miles of course uses fuel and we**

**//would now find less fuel in the tank.**

myBmw.takeTrip(100);

**//Use the *myBmw* object to call the *reportFuel* method. It returns a**

**//double value of the amount of gas left in the tank and this is assigned**

**// to the variable *fuel\_left***

double fuel\_left = myBmw.reportFuel( );

**//Print the *fuel\_left* variable**

System.out.println(fuel\_left); //prints gallons left, **15.833333333333332**

}

}