**Lab 03 – BigMacWalk**

Open BlueJ, and create a new BlueJ project titled **Lab03-BigMacWalk** in your CS\LABS folder (H:\CS\Labs).

Create a new class with this code skeleton:

//Name:



public class PracticeProblems

{

public static void main(String[] args)

{

}

}

Computers were made to compute – in this lab, you will get more practice using print statements, variables and arithmetic operations. For now, the value of all variables are ‘hard coded’ – meaning they have values defined by you, the programmer. A better version of these programs might take the values of the variables from the keyboard (user) – we’ll learn how to do that shortly!

**Before each problem, insert a COMMENT with the problem number.**

1. Declare a double variable called tuna, and set it equal to 2.99. Write the code to print the value of the tuna variable plus 3. (In this first problem, you shouldn’t change the value of tuna. Just print to the screen what tuna+3 is.)
2. Add one to the value of the tuna variable using the increment operator (++), then print the new value of tuna*.*
3. Add five to the value of tuna (without using the ++ operator). Print the new value of tuna.

Data flows right to left. To add 5 to the value of a variable, we say the new value of the variable is equal to the old value of the variable plus 5. Or, remember that there is an assignment operator that can also add.

1. Change the value of tunato equal tuna divided by 10. Print the new value to the screen.
2. Declare two integer variables a and b*.* Set them equal to 4 and 6. Write the code to calculate (and print):

2 \* a + 3 \* b \* (a + b) / 2

1. Declare two integer variables c and d*.* Set them equal to 5 and 9. Write the code to calculate (and print):

-1 \* (c + d) \* (c + d) / 4

1. (Riddle) What’s a 7-letter word that can contain more than 7 letters?
2. Create a new double variable called something, equal to 4.5. Write the code to print the value of something squared. **Don't use the ^ operator! Refer to the notes for help.**
3. Create two doubles, PI and radius, equal to 3.14159 and 6.5. (Will the value of pi ever change? Remember that unchanging variables should be *final*.) Write the code to print the area of this circle. Reassign the value of radius to 7.2 and print the area of the new circle.
4. Create two integer variables, cards and players, equal to 52 and 6. Your goal is to deal out as many of the cards as you can so that each player gets the same number of cards. How many will each player get? How many cards will be left over? Print this information to the screen. (Remember modulus?)
5. Complete the ‘Worksheet – Data Types and Math’ (Excel file). Make sure to save it when done.

**Big Mac Walk**

In your current project, create a new class with this code skeleton:

//Name: 

public class BigMacWalk

{

public static void main(String[] args)

{

}

}

1. Declare an integer variable called numBigMacs*,* and assign it the number 4*.* This is the number of Big Macs you will potentially eat this week.
2. Given that a Big Mac has **590 calories** and that you (on average) burn **94 calories per mile** walked, write a program that will print out the total number of calories consumed (given the number of Big Macs eaten) and the number of miles you need to walk to burn off those Big Macs. Make sure your output matches the sample output.

/\* Hint – create 2 additional variables of type double that would hold this important information: double calsPerMile = 94 and double bigMacCals = 590 \*/

NOTE: For testing, change numBigMacs to something other than 4 to see if all the calculations still work correctly to give the new appropriate numbers.

1. The users of our app are very hungry. With every Big Mac, they also had fries and a drink. This adds an extra 400 calories per meal. (Add a variable called friesAndDrinkCals with a value of 400.) How many miles they would need to walk to burn off all those Big Mac meals. Make sure your output matches the sample output.