**Name:**

**Advanced Programming in Java**

**Lab Exercise 9/3/2024**

In this exercise, you will learn about what can happen if you mix data types. You will also learn about “type casting”.

1. Read Lesson 5 in Blue Pelican.

Unless otherwise instructed in the following problems, state what gets printed.

1. Write code that will create a constant *E* that’s equal to 2.718.

2. Write the simplest type constant that sets the number of students, *NUM\_STUDENTS*,

to 236.

3. What’s wrong with the following code in the *main* method?

final double Area;

Area = 203.49;

4. int cnt = 27.2;

System.out.println(cnt);

What’s printed?

5. double d = 78.1;

int fg = (int)d;

System.out.println(fg);

What’s printed?

6. Is *double f4 = 22;* legal?

7. The following code stores a 20 in the variable *j*:

double j = 61/3;

What small change can you make to this single line of code to make it print the “real”

answer to the division?

8. System.out.println( (double)(90/9) );

9. System.out.println(4 + 6.0/4 + 5 \* 3 – 3);

10. int p = 3;

double d = 10.3;

int j = (int)5.9;

System.out.println(p + p \* d – 3 \* j);

11. int p = 3;

double d = 10.3;

int j = (int)5.9;

System.out.println(p + p \* (int)d – 3 \* j);

The following code applies to 12 – 15:

int dividend = 12, divisor = 4, quotient = 0, remainder = 0;

int dividend2 = 13, divisor2 = 3, quotient2 = 0, remainder2 = 0;

quotient = dividend/divisor;

remainder = dividend % divisor;

quotient2 = dividend2 / divisor2;

remainder2 = dividend2 % divisor2;

12. System.out.println(quotient);

13. System.out.println(remainder);

14. System.out.println(quotient2);

15. System.out.println(remainder2);

16. Write a line of code in which you divide the double precision number *d* by an integer

variable called *i*. Type cast the *double* so that strictly integer division is done. Store

the result in *j*, an integer.

17. Suppose we have a line of code that says

final String M = “ugg”;

Later in the same program, would it be permissible to say the following?

M = “wow”;

Now for something different; you will learn how to perform input and output operations using the JOptionPane. Here is an example of the JOptionPane for input and output

Output

JOptionPane.showMessageDialog(null, "alert", "alert", JOptionPane.ERROR\_MESSAGE);

Note: The last two parameters are optional.

Input

String inputValue = JOptionPane.showInputDialog("Please input a value");

Note: The input is treated as a String and must be converted to the appropriate data type. For example if the inputValue needed to be converted into an integer, you could use the following:

int inputNumber = Integer.parseInt(intputValue);

1. Write a class Calculate that inputs three integers from the user and displays and displays the sum. Write this program using a JOptionPane. Here is a skeleton of your program

// Perform simple calculations on three integers.

import javax.swing.JOptionPane;

public class Calculate

{

public static void main( String[] args)

{

String firstNumber; // first string entered by user

String secondNumber; // second string entered by user

String thirdNumber; // third string entered by user

int number1; // first number

int number2; // second number

int number3; // third number

int sum; // sum of the numbers

/\* write a series of statements to read in three numbers, convert them

to integers, and assign them to number1, number2, and number3 \*/

// perform calculations

sum = number1 + number2 + number3;

// create result string

String result;

/\* Write a statement that concatenates all the results into a single string that

shows the result such as 12 + 7 + 8 = 27 \*/

// display results

JOptionPane.showMessageDialog( null, result, "Calculation Results",

JOptionPane.INFORMATION\_MESSAGE );

System.exit( 0 );

}

}

**Project… Mixed Results**

Create a new project called *MixedResults* with a class called *Tester*. Within the *main* method of *Tester* you will eventually printout the result of the following problems. However, you should first calculate by hand what you expect the answers to be. For example, in the parenthesis of the first problem, you should realize that strictly integer arithmetic is taking place that results in a value of 0 for the parenthesis.

double d1 = 37.9; //Initialize these variables at the top of your program

double d2 = 1004.128;

int i1 = 12;

int i2 = 18;

Problem 1: 57.2 \* (i1 / i2) +1

Problem 2: 57.2 \* ( (double)i1 / i2 ) + 1

Problem 3: 15 – i1 \* ( d1 \* 3) + 4

Problem 4: 15 – i1 \* (int)( d1 \* 3) + 4

Problem 5: 15 – i1 \* ( (int)d1 \* 3) + 4

Your printout should look like the following:

Problem 1: 1.0

Problem 2: 39.13333333333333

Problem 3: -1345.39999999999

Problem 4: -1337

Problem 5: -1313

**When you complete this project, submit your documented source code.**