Practice 3: Data & Operators*

Kenneth Leroy Busbee

This work is produced by OpenStax-CNX and licensed under the
Creative Commons Attribution License 3.0†

Abstract

Questions, exercises, problems, etc. that support this chapter in the "Programming Fundamentals - A Modular Structured Approach using C++" collection/textbook.

1 Learning Objectives

With 100% accuracy during a: memory building activity, exercises, lab assignment, problems, or timed quiz/exam; the student is expected to:

1. Define the terms on the definitions as listed in the modules associated with this chapter.
2. Understand basic data types and how operators manipulate data.
3. Given pseudocode and test data documents, write the C++ code for a program

2 Memory Building Activities

Link to: MBA 03[1]

3 Exercises

Exercise 1 ***(Solution on p. 4.)***

Answer the following statements as either true or false:

1. A data type defines a set of values and the set of operations that can be applied on those values.
2. Reserved or key words can be used as identifier names.
3. The concept of precedence says that some operators (like multiplication and division) are to be executed before other operators (like addition and subtraction).
4. An operator that needs two operands, will promote one of the operands as needed to make both operands be of the same data type.
5. Parentheses change the precedence of operators.

*Version 1.4: Mar 22, 2010 9:05 pm +0000

†http://creativecommons.org/licenses/by/3.0/

[1]See the file at <http://cnx.org/content/m22455/latest/index.html>

4 Miscellaneous Items

Link to: Manipulation of Data Part 1[2]

5 Lab Assignment

5.1 Creating a Folder or Sub-Folder for Chapter 03 Files

Depending on your compiler/IDE, you should decide where to download and store source code files for processing. Prudence dictates that you create these folders as needed prior to downloading source code files. A suggested sub-folder for the **Bloodshed Dev-C++ 5 compiler/IDE** might be named:

- Chapter_03 within the folder named: Cpp_Source_Code_Files

If you have not done so, please create the folder(s) and/or sub-folder(s) as appropriate.

5.2 Download the Lab File(s)

Download and store the following file(s) to your storage device in the appropriate folder(s). You may need to right click on the link and select "Save Target As" in order to download the file.
Download from Connexions: Solution_Lab_03_Pseudocode.txt[3]
Download from Connexions: Solution_Lab_03_Test_Data.txt[4]

5.3 Detailed Lab Instructions

Read and follow the directions below carefully, and perform the steps in the order listed.

- Navigate to your sub-folder: Chapter_03 and open and study the two files.
- We have learned that a fundamental concept of interaction with computers is to divide the problem/task into three parts – input, processing and output. This problem is simple and we will use the IPO (input – processing – output) approach again. **However this time we are going to think about it backwards.**

- **What output do I want displayed?**
- Number of gallons of paint
- Total cost of the paint

- **Thus, what calculations do I need to make?**
- Total cost of the paint is the Number of gallons needed times price per gallon
- Number of gallons needed is the Total area to be covered (let's use square feet) divided by the coverage per gallon of paint (Note: you must round up to the next full gallon of paint.)
- Total area to be covered is the Length times height times 2 added to the width times height times 2

- **Which leads us to, what data do I need as input?**
- Price of a gallon of paint
- Number of square feet that a gallon will cover

[2]See the file at <http://cnx.org/content/m22455/latest/Manipulation_Data_Part_1.pdf>
[3]See the file at <http://cnx.org/content/m22455/latest/Solution_Lab_03_Pseudocode.txt>
[4]See the file at <http://cnx.org/content/m22455/latest/Solution_Lab_03_Test_Data.txt>

- Length of the house
- Width of the house
- Height of the house

- You can see that by working the logic backwards, we can start to completely see what the program must do. We need to enter some data (input), do some calculations (process) and display the results (output).
- Copy into your sub-folder: Chapter_03 one of the source code listings that we have used (we suggest the Lab 01 source code) and rename the copy to: **Lab_03.cpp**
- Modify the code to follow the Solution_Lab_03_Pseudocode.txt file.
- I am just going to give you the line of code for rounding up to the next whole gallon of paint (See the "Data Type Conversions" module within Chapter 3 of the Connexions materials. Do you understand why it works?).
- **total_gal_paint = total_area / coverage_gal_paint + 0.9999;**
- Build (compile and run) your program. You have successfully written this program when it runs with your test data and gives the predicted results.

- After you have successfully written this program, if you are taking this course for college credit, follow the instructions from your professor/instructor for submitting it for grading.

6 Problems

6.1 Problem 03a – Instructions

Write the C++ code to do the following pseudocode example.

Example 1: pseudocode

```
Prompt the user for his monthly income.
Get the users monthly income.
Multiply the monthly income by 12.
Display the annual income.
Pause the program so the user can see the answer. (HINT: You may call a function.)
```

Solutions to Exercises in this Module

Solution to Exercise (p. 1)

Answers:

1. true
2. false
3. true
4. true
5. false – Parentheses change the order of evaluation in an expression.