

CMPE – 211

Preliminary Work (Pre-Lab Activity)

Laboratory Experiment #1

Textbook Material:	Chapter 1	«Elementary C++ Programming»	pp.1-15
	Chapter 2	«Fundamental Types»	pp.16-35
	Chapter 3	«Selection»	pp.36-59
	Chapter 4	«Iteration»	pp.60-74

••• Tutorial

The Tutorial covers introduction to Microsoft (MS) Visual C++ 6 IDE, which is used in all following laboratory experiments. Tutorial is conducted by one of the Course Lab. assistants who guides Your laboratory group. Explanations are based on the «Dive into Microsoft Visual C++ 6» (24 pages, previously known as «Getting Started with MS Visual C++ 6») publication from Deitel and Associates, Inc. (<http://www.deitel.com/books/downloads.html>).

Main topics to cover during 1-hour tutorial:

- how to use MS Visual C++ Integrated Development Environment (IDE),
- how to create, compile and execute C++ console application,
- how to use debugger to locate logical errors in programs,
- how to use Help facilities and MSDN Library.

All these skills are essential for performing successfully upcoming lab tasks.

Read «Dive into Microsoft Visual C++ 6» publication in advance (before entering the Lab), even if You already took the course «Data Structures» before. The electronic copy of this publication can be also found at http://www.deitel.com/books/cppHTP3/cppHTP3_GSW_vcpp6.pdf (360 Kb).

••• Example

☀ Write (copy), Modify, Compile, Execute and Debug C++ program covered by the following example:

Example

[file lab_tut1.cpp]

(see also MSDN Library on using parameterized (header `iomanip.h`) and non-parameterized manipulators – these are **fixed**, **scientific**, **endl**, **setprecision**. Understand their meaning and the way to use them in C++ programs. Try in practice other appropriate manipulators as well)

MSDN : «Except for macro names, ... all names in the Standard C++ Library are declared in the **std namespace**»

```
// file lab_tut1.cpp – Visual C++ 6 (Introductory Edition)
#include <iostream>
#include <iomanip>
using namespace std;
int main()
{
    const double i = 1.3;
    double j;
    cout << "Enter a number for calculations: ";
    cin >> j;
    cout << "Default output form:" << i << " " << j << endl;
    cout << fixed << setprecision(2) << "Result: ";
    cout << i << " multiplied by " << j << " is ";
    cout << setprecision(4) << i*j << scientific << " (or, appr.>";
    cout << setprecision(3) << i*j << ")" << endl;
    return 0;
}
```

• • • TASK 1

Write, Compile and Execute a C++ program that finds the **largest** of several integers. The program asks for the **number of integers to process** [to be entered from the keyboard]. Afterwards, the program finds the largest of the entered integer sequence. Found value is printed.

- **NOTE:** For this and all following tasks you are free to use any basic constructions and data types (*arithmetic*, i.e. integral and floating-point; *compound*, e.g. arrays) of C++. We assume that a core of these topics is already known from previous C language-based courses (e.g. «Introduction to Programming» and/or «Data Structures Using C»).

• • • TASK 2

Write, Compile and Execute a C++ program that calculates and prints (a) **the product of the integers**, and (b) their **average**. The program reads an integer and **checks whether it is zero or not**. If the number is zero, it is discarded. After each number entry, the program asks a user whether he/she wants to continue. If the answer is **YES** (processed as a character 'Y'), it asks for the next input integer and continues this way until user's answer is **NO** (processed as character 'N'). As a result, the program calculates the product of all entered **non-zero** integers and their average. The program's output has the following format:

In total *<value1>* numbers are processed
The product of non-zero numbers is *<value2>*
The average is *<value3>*

- **NOTE:** Use C++ standard Boolean type **bool** for processing values 'Y'/'N'. Try both **for** and **while** loops in the program. The product value appears in the output with 3 digits after the decimal dot, i.e. as `___ .zzz`, where *z* stands for a single digit.

• • • Appendix

- Check **Review Questions** at the end of textbook's **Chapters 1-4** (pp. **9, 32, 49** and **80**),
- Use **debugging** facilities of the Visual C++ compiler while preparing programs for TASKS 1-2 (follow tutorial «Dive Into Microsoft Visual C++ 6» by *Deitel & Associates*, Lab Experiment #0). You are free to use any other C++ compiler in your work.

• • • Sources

- *John R. Hubbard*. Schaum's Outline of Programming with C++, 2nd edition, McGraw-Hill, 422 p., 2000
- *Harvey M. Deitel, Paul J. Deitel*. C++ How To Program, 4th edition, Prentice Hall, 1320 p., 2002
- C++ in the Lab : Lab Manual to accompany C++ How To Program, 4th edition