

Exercise One – Introduction to Microsoft Visual C++ .Net 2003

Larry Caretto
Computer Science 106
Computing in Engineering and Science

January 31, 2006

California State University
Northridge

Overview

- Using Visual C++ Integrated Development Environment (IDE)
 - Commercial grade system
 - May be confusing to beginning programmers
 - Will give you experience with real-world IDE
- C++ programs used today have many elements that you will not see until later
- Important objective is learning use of IDE to enter, edit, compile and debug code

California State University
Northridge

Items to accomplish today

- Get account number
- Go to www.csun.edu/~lcaretto/comp106
 - Download exercise one from “What’s New” section of home page
- Follow instructions handed out in class to start IDE and run first exercise
 - Define project and source code file
 - Cut and paste source code from assignment
 - Compile, execute, check results, and revise
 - Submit printed output for tasks three and four

California State University
Northridge

Four Documents for This Lab

- Handed out in class
 - Account number instructions
 - Separate account for College network with same user ID and password as CSUN account
 - Instructions for use of the IDE
 - Introduction for today and reference for later
- From www.csun.edu/~lcaretto/comp106
 - Download exercise one instructions from “What’s New” section of home page
 - This slide presentation
- Lecture notes for today and Thursday

California State University
Northridge

Getting Started

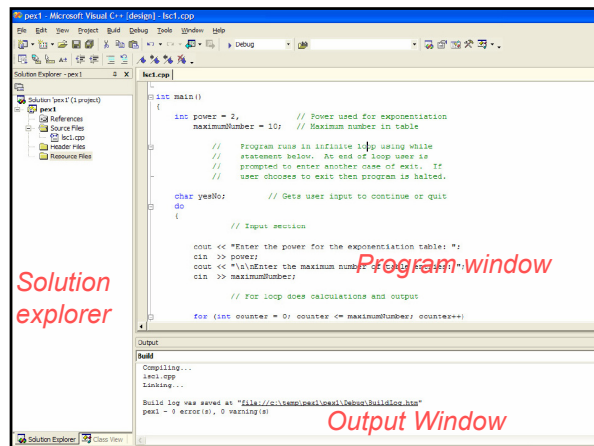
- Use both IDE instructions handed out in class and online exercise notes
- Start with IDE notes to learn how to define project and get blank source code file
- Copy code from exercise notes into file
- Follow IDE notes to compile, link and execute program
- Try to solve problems before asking for help

California State University
Northridge

The IDE

- Defines project which can have many files
 - Will only use one file per project for all but one programming assignment
- Must have correct identification of source file with project
 - May have multiple files open for editing
 - May not list correct program file for project
- One IDE window, known as solution explorer, shows source files for project
 - See next chart

California State University
Northridge



Four Tasks in Exercise One

- Print tables of numbers raised to a power
 - Task one: copy, paste and execute code that prints the squares of numbers from 0 to 10
 - Task two: same actions for modified code with user input of power and number of entries
 - Task three: new version repeats code execution as long as user wants
 - Copy, paste, execute and copy comments into code
 - Use best guess of correct location for comment
 - Task four: same version as three with errors
 - Copy, paste and see compiler syntax error messages

California State University
Northridge

Questions for Task Three

- Repeat calculations that produce tables of numbers from 0 to N raised to a power, p
- You are asked to identify the parts of the code that do the following
 - Reads the power p from the keyboard
 - Prints the output to the screen
 - Repeats calculations for each line of the table
 - Repeats the entire calculation for new input
- Copy comments from exercise to code

California State University
Northridge

Using Comments

- Comments in a code explain code operation to human reader
- Create comment by starting with `//`

```
// this is an example of a comment
x = y; // set x = y (useless comment)
// line below starts iteration
for ( int i = 0, i < MAX; i++ )
```
- IDE colors show keywords and comments

California State University
Northridge

Syntax errors

- Compiler checks for errors in the use of the C++ language called syntax errors
- Version four is really version three edited to introduce syntax errors
- Run version four and see syntax error messages at bottom of screen
- Click on line number of error message to show a blue arrow pointed to line where the compiler detected the error

California State University
Northridge

Correcting Syntax Errors

- Read error message and locate line where error is said to occur
- Error messages are not always correct and may refer to adjacent lines
- After your run task four code
 - Find line where error is reported
 - Locate and correct error with reference to task three code if necessary
 - Correct code as indicated on next slide to show error and correction

California State University
Northridge

Correcting Syntax Errors II

- Technique used here is done to record corrected errors during development
- Will be used when you email code to me for help
- Statement with error: `int x = 2`
- Correction that records error in **original**
`//int x = 2` missing ; correct below
`x = 2;`
- `//` before statement makes it a comment

California State University
Northridge

Some Final Observations

- Each statement (which may be over more than one line) ends in a semicolon (;)
- In several places curly braces, { and }, appear. These set off logical blocks of code subject to program controls
- The command `cout` with the insertion (`<<`) operator is used for screen output and the command `cin` with the extraction (`>>`) operator is used for keyboard input.

California State University
Northridge

General Program Template

```
#include <iostream>
using namespace std;
int main()
{
    // Put your program statements in
    // place of this comment

    return EXIT_SUCCESS;
}
```

California State University
Northridge