Study recommendations for Quiz A

Review Chapters 1 - 3 in your textbook.

Review the following web pages about:

Overview at http://www.gibsonr.com/classes/cop2000/probsolv.html

Data Types & Language at http://www.gibsonr.com/classes/basics/datatypes.html

Data Types in C++ at http://www.gibsonr.com/classes/cop2000/datatypesc.html

Chapter 1-3 section of the Glossary at http://www.gibsonr.com/classes/cop2000/glossary.html

Primary Problem Solving Example at http://www.gibsonr.com/classes/cop2000/example prob solv.html

Constants and Symbolic Constants in C++ Programming at

http://www.gibsonr.com/classes/cop2000/constants.html

Variables in C++ Programming at http://www.gibsonr.com/classes/cop2000/variables.html

Expressions & Operators in C++ at http://www.gibsonr.com/classes/cop2000/expressions.html

Order of Precedence of Operators in C++ at http://www.gibsonr.com/classes/cop2000/precedence.html

Acquiring Input in C++ at http://www.gibsonr.com/classes/cop2000/input.html

Rounding Numbers in C++ at http://www.gibsonr.com/classes/cop2000/rounding.html

Output Formatting in C++ at http://www.gibsonr.com/classes/cop2000/output formatting.html

Displaying Special Characters in C++ at http://www.gibsonr.com/classes/cop2000/tip dispec.html

Remember the Following

Standard problem solving steps used in program development (in order):

ANALYSIS: LOGICAL DESK WORK

- 1. Define the Task or Objectives
- 2. Define the Output
- 3. Define the Input
- 4. Define the Algorithm
- 5. Check the Algorithm

PROGRAMMING: PHYSICAL CODING

- 6. Code the Program in a high-level language such as C++
- 7. Debug the Program

The rules of syntax for defining valid <u>identifiers</u> in the C++ Language:

- Identifiers can contain only letters (either case), numerals, or the underscore character (_), although identifiers starting with an underscore had special restrictions and an identifier consisting of only a lone underscore is forbidden.
- No blank spaces or special characters are allowed; nor should you use double underscores.
- Identifiers cannot start with a digit.
- A C++ reserved word cannot be used as an identifier.
- Identifiers are case sensitive.
- The maximum length of an identifiers depends on the compiler being used, but most allow at least 1024 characters (although such lengths would be extremely impractical).

Rev. 2011-9-5 Page 1

Test preparation exercise:

Circle the labels that are NOT valid variable names under the rules of syntax for C++.

cout

xYz2









my2nd

Note that <u>cout</u> is a <u>standard identifier</u> in C++ but not a <u>reserved word</u>, like <u>int</u>.

The rules defining the hierarchy of operators when evaluating expressions in C++:

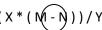
- Operators in parentheses are done first, from innermost to outermost.
- *, /, and % are done next, as encountered from left to right.
- + and are done next, as encountered from left to right.

Test preparation exercise:

In <u>each</u> of the four separate expressions below, circle the <u>operator</u> of the arithmetic operation that will be performed first.



X - (*)



Z / ((- Y))

You will be tested on your ability to translate steps from an algorithm into C++ statements. Study the examples of C++ code given to you in this class and give careful attention to the syntax of each statement.

Test preparation exercise:

For each of the following logical statements, write <u>one</u> C++ statement to perform the specified task. Assume that these tasks are all a part of a restaurant-bill-calculating program <u>using identifiers as shown below</u>:

LABEL	DESCRIPTION	SOURCE	USAGE	DESTINATION
PUR	Amount of the purchase	Keyboard	for TAX & PAY	
TAX	Calculated sales tax	Calculation	for PAY	
PAY	Total amount to pay after tax	Calculation	-	Displayed

1. Allocate (set aside) storage for the three variables that may have decimal points.

double PUR,TAX,PAY; // Note: the float data type would also be OK for small values

2. Display a prompt asking the user to enter the amount of the purchase.

cout << "Amount purchased? ";</pre>

3. Store the user's keyed-in answer into the variable that holds the purchase.

cin >> PUR;

4. Calculate and store the tax as 6% of the purchase amount.

TAX = 0.06 * PUR;

5. Display the string "Please pay \$" followed by the total amount to pay rounded to cents, followed by a carriage return.

cout << "Please pay \$" << setprecision(2) << fixed << PAY << endl;

Rev. 2011-9-5 Page 2

You will be tested on your ability to recognize proper C++ syntax. Study the C++ code in your textbook and in the sample below and give careful attention to the structure and grammar of each statement. Remember that it is the punctuation that is important in C++, not the layout of the source code.

```
/* demo.cpp - A demonstration Program */
/* Written by R. Gibson - 9/1/2011 */
#include <iostream>
                      // for cout and cin
using namespace std; // to define context for cin and cout
#include <iomanip>
                       // for setprecision and fixed
#define RATE 0.06
                       // Symbolic constant for the tax rate
int main (void)
 double AMT,TAX,PAY;
 cout << "Amount of sale? ";
 cin >> AMT;
 TAX = AMT * RATE;
 PAY = AMT + TAX;
 cout << "Please pay $" << setprecision(2) << fixed << PAY << endl;</pre>
 return 0;
}
```

Test preparation exercise:

Circle any (and all) SYNTAX errors in the following C++ statements and describe them briefly on the line provided. Do NOT mention logical errors. If a line is valid, answer "OK". Each line is a separate question, independent of the others.

#include <iostream>;</iostream>	Compiler directives are not terminated by a semi-colon		
#define FACTOR 😏	Assignment is not used with a symbolic constant		
float (=)AMT;	Equal sign can be used only after an identifier to initialize		
X = X * 2;	OK (this statement simply doubles X)		
cout < Pello, // Display the word Hello	String constants should be enclosed in double quotes		
Float AMT,PAY,TAX=10.0;	The keyword float is all lowercase		
/ This is a commen /	Comments must be inside /* and */ pairs or preceded by //		
cin AMT; // Read AMT from keyboard	cin should use the >> stream extraction operator		
NUM = 100 / (AMT + 5 ()	The statement terminating semi-colon is missing		
cout << "Hello("\n;)	The \n escape sequence should be inside the quotes		

Rev. 2011-9-5 Page 3