# CS 2275 Exercise 2 – Basic C++

Module 2 Objectives:

1. Variable types assignment statements, casting, literals, booleans,
2. loops, if then else, switch, if expressions
3. functions, parameters, simple pointers
4. other weird C++ stuff (e.g., break, continue, ++i, i++
5. Testing function correctness.

Readings for Module 2: Lippman - 1.4,2.1-2.5,4.1-4.7, 4.9, 4.11, 4.12, 5.1-5.5, 6.1-6.3

Using Dev-C++, create a .cpp source file titled ex2-<your last name>.cpp that contains solutions to the following problems. Use a .doc or .txt file ex2<your last name>.doc to provide the answers to the text questions. As with all assignments in this course, Be sure to include a block comment just about the function indicating its purpose and use a really good function name. Include code under main to test your function by allowing the user to enter the needed parameters. Make sure you test your function sufficiently to insure correctness.

1. Write a C++ function that accepts two integers n1 and n2 and returns the sum of all the integers between n1 and n2 inclusive (e.g. including both n1 and n2). Do not assume that n1 is greater than or less than n2. Include code under main to test your function by allowing the user to enter values for n1 and n2.
   1. What was your test plan for this function that you used to determine it worked correctly?
2. Write a C++ function that accepts an unsigned integer year and returns true if and only if (IFF) that year is a leap year. According to the Gregorian calendar, leap years are evenly divisible by 4 (e.g. 1996 IS a leap year) except when the year is evenly divisible by 100 (e.g. 1900 was NOT a leap year) except when the year is evenly divisible by 400 (e.g. 2000 WAS a leap year). An elegant solution for this function will only have a single Boolean expression in the return statement. Include code under main to test your function by allowing the user to enter values for n1 and n2.
   1. What was your test plan for this function that you used to determine it worked correctly?
3. Write a void function that accepts an integer n and uses cout statements to display the following pattern in the console window (for n = 4). Be sure to include a block comment just about the function indicating its purpose and use a really good function name. Include code under main to test your function by allowing the user to enter a value for n.

---\*

--\*\*

-\*\*\*

\*\*\*\*

1. The following function generates the corresponding character for an asci value between 0 and 255. Use it to determine the asci values for 45 and 97. Place your results in a .doc (or .txt) file named ex2-<your last name>.doc. there may be a syntax error you need to fix.  
     
   char getCharFromAsciiValue (int n) {

return (static\_cast<char> (n));

}  
what happens if you remove the static\_cast from the return statement? Why?

1. What will the following C++ code display in the console window? Place your answer in the .doc file for this assignment.

int main() {

int n =5;

int \*p = &n;

\*p++;

cout << n<< endl;

return 0;

}

1. Write a function sumDigits that accepts a positive integer n and returns the sum of the digits in that integer. Use math, not strings! Return -1 if the integer is negative. Thus sumDigits(1234) would return 10, sumDigits(-34) would return -1, and sumDigits(42) would return 6. Include code under main to test your function by allowing the user to enter a value for n
2. Using paper and pencil, determine what parameter n for the following function will return 42. Place your results in the .doc (or .txt) file named ex2-<your last name>.doc  
   int foo(int n) {

int a = n, b = n+a, sum = 0;

while (a<b) {

a++;

b--;

sum = a+b;

}

return sum;

}

Submit your .cpp and .doc files using moodle

Grading rubric: Problems 1 and 2: 15 points problems 3 through 7 : 14 points. Style: up to - 10 points -- poor variable or poor function names or no block comment with a goal statement for a function. Solutions that have significantly more lines that needed will be docked points for lack of elegance.

**REMINDERS**

* code that does not compile will not be graded
* the grader should not need to modify/uncomment your code to test it. Provide a test mechanism allowing the grader to enter various tests for teach function.