# CS2275 Exercise 12 – Shifts, Masks, Packing and Unpacking

Module 12 objectives: shifts and other bitwise operators, masking, packing and unpacking, struts

Module 12 Readings: Lippman -4.8 page 141

Using Dev-C++, create a .cpp source file titled ex12-<your last name>.cpp that contains the C++ solutions for the problems listed below. Also create a .doc file containing text solutions.

1. Write C++ code that creates 4 char variables (c1, c2, c3, c4) containing positive integers <=15 (e.g. 3, 5, 7, 9). Then write code that creates an unsigned 16bit short packed1 and use shifts and the bit wise or operator ( | ) to pack c1 into bits 0-3 of packed1, c2 into bits 4-7, c3 into bits 8-11, and c4 into bits 12-15. Use the bitset gadget to make sure everything works correctly.
2. In the same .cpp file used for problem 1, create 4 additional char variables (c5, c6, c76, c8) create additional C++ code that uses shifts, masks, and the bitwise AND operator ( & ) to unpack packed1 each of the 4 bit items in packed1 back into the original numbers (e.g. bits 0-3 into c5, bits 4-7 into c6, etc.
3. Using the example functions in note 12.5 as a template, write a function packCharsIntoInt that accepts 4 char variables and returns a single unsigned32bit int that has the first character packed into bits 0-7, the second into bits 8-15, the third into the third byte, and the fourth into the top byte. Use bitset to output the results to insure things work correctly.
4. Write a C++ function unpackCharsFromInt that accepts a single unsigned int by reference and has 4 char parameters passed by reference that unpacks the bytes from the int and puts them into the char parameters. Use packCharsIntoInt and unpackCharsFromInt to test each other.

Grading: 20 points each. Style – 10: 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.