## Jeremy Rios-Martinez

## COMSC 265

## 8/20/2010

## Lecture Review and Lab 2 Assignment

**Exercise 1**: Write two versions of a 10 line or less program, the first program that uses magic numbers in the condition of an “if” statement, the second program that replaces it with Named Numeric Constants.

**int main(void) //** version 1

**{**

**int customerAge;**

**cout << "Enter age of customer: ";**

**cin >> customerAge;**

**if (customerAge < 18)**

**{**

**cout << "Im sorry but you are not old enough to purchase tobaco products" << endl;**

**}**

**else**

**{**

**cout << "Thank you for you purchase and enjoy lung cancer " << endl;**

**}**

**system ("pause");**

**return 0;**

**}**

**#define LEGAL\_AGE 18**

**int main(void) //** version 2

**{**

**int customerAge;**

**cout << "Enter age of customer: ";**

**cin >> customerAge;**

**if (customerAge < LEGAL\_AGE) //** LEGAL\_AGE = 18

**{**

**cout << "Im sorry but you are not old enough to purchase tobaco products" << endl;**

**}**

**else**

**{**

**cout << "Thank you for you purchase and enjoy lung cancer " << endl;**

**}**

**system ("pause");**

**return 0;**

**}**

**Exercise 2**: Write two versions of a 15 line or less program, a first program that uses non Boolean relational conditions in an “While” loop statement, the second program that replaces relational term with a single Boolean variable.

**#define LEGAL\_AGE 18**

**#define EXIT 1**

**int main(void)** //version 1

**{**

**int customerAge,loop = 0;**

**while (loop != EXIT) // EXIT = 1**

**{**

**cout << "Enter age of customer: ";**

**cin >> customerAge;**

**if (customerAge < LEGAL\_AGE) // LEGAL\_AGE = 18**

**{**

**cout << "Im sorry but you are not old enough to purchase tobaco products" << endl;**

**}**

**else**

**{**

**cout << "Thank you for you purchase and enjoy lung cancer " << endl;**

**}**

**cout >> "enter the number 1 to exit the program or enter the number 2 to run again: ";**

**cin >> loop;**

**}**

**system ("pause");**

**return 0;**

**}**

**#define LEGAL\_AGE 18**

**int main(void)** //version 2

**{**

**int customerAge, exit;**

**bool loop = true;**

**while (loop)**

**{**

**cout << "Enter age of customer: ";**

**cin >> customerAge;**

**if (customerAge < LEGAL\_AGE) // LEGAL\_AGE = 18**

**{**

**cout << "Im sorry but you are not old enough to purchase tobaco products" << endl;**

**}**

**else**

**{**

**cout << "Thank you for you purchase and enjoy lung cancer " << endl;**

**}**

**cout >> "enter the number 1 to exit the program or enter the number 2 to run again: ";**

**if (exit == 1)**

**{**

**loop = false;**

**}**

**}**

**system ("pause");**

**return 0;**

**}**

**Exercise 3**: Complete the right two columns in the table below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **(NOT A) OR B** | **A OR (B AND (NOT C))** |
| **T** | **T** | **T** | T | T |
| **T** | **T** | **F** | T | T |
| **T** | **F** | **T** | F | T |
| **T** | **F** | **F** | F | T |
| **F** | **T** | **T** | T | F |
| **F** | **T** | **F** | T | T |
| **F** | **F** | **T** | T | F |
| **F** | **F** | **F** | T | F |

**Exercise 4**: Write two versions of a 15 line or less program, one that uses one version of a condition conditions in an “if else” statement, the second that replaces it with equivalent but shorter condition.

**int main(void) //** version 1

**{**

**bool A = true, B = true, C = true;**

**if ((A && !B) || (A && !(B || C)) || (B && !(B || C)))**

**{**

**Do some thing;**

**}**

**system ("pause");**

**return 0;**

**}**

**int main(void) //** version 2

**{**

**bool A = true, B = true, C = true;**

**if (A && !B) // (A && !B) || (A && !(B || C)) || (B && !(B || C))**

**{ // (A\*!B) + A(!B\*!C) + B(!B\*!C)**

**Do some thing; // (A\*!B) + (A\*!B\*!C) + (0\*!C)**

**} // (A\*!B)(1+!C) + 0**

**system ("pause"); // (A\*!B)1**

**return 0; // A && !B**

**}**

**Exercise 4**: Write a 25 line or less program, has a compound conditions in an “if” statement. Use 4 expressions in the conditions. Place the expressions in the best order of execution. Explain why they are in that order.

Note: Expressions in a compound condition should be arranged in order of expected frequency

**#define TARGET 13**

**int main(void)**

**{**

**int answer;**

**bool loop = true;**

**cout << "Im thinking of a number between 1 and 50 see if you can guess it: ";**

**while (loop)**

**{**

**cin >> answer;**

**if (answer > TARGET || answer < TARGET) // conditions are in the order**

**{ // they are in becaue it is more**

**if (answer > TARGET) // likley answer guessed will be**

**{ // will be greater than TARGET**

**cout << "To High... Guess again: ";**

**}**

**else**

**{**

**cout << "To Low... Guess again: ";**

**}**

**}**

**else**

**{**

**cout << "Wow you guess it. You must be psychic\n";**

**loop = false;**

**}**

**}**

**system ("pause");**

**return 0;**

**}**

**Exercise 5**: Write two versions of a 10 line or less program, one that uses non Boolean conditions in an “if else” statement, the second that replaces it with Booleans

**int main(void) //** version 1

**{**

**int A = 1, B = 5, C;**

**if (C < A && < B)**

**{**

**Do some thing;**

**}**

**else**

**{**

**Do something else.**

**}**

**system ("pause");**

**return 0;**

**}**

**int main(void) //** version 2

**{**

**bool A = true;**

**if (A)**

**{**

**Do some thing;**

**}**

**else**

**{**

**Do something else.**

**}**

**system ("pause");**

**return 0;**

**}**

**Exercise 6**: What is de Morgan’s Law. Give an example when you would use it. What other classes might you take to help you with conditions in Decision structures and Repetition Structures?

1. De Morgan’s is a set of laws that allow programmers to transform long complicated conditions into shorter much simpler equations that have the same outcome.
2. System logistics and Boolean algebra.