**COSC 359**

**String Encryption and Decryption**

Strings are stored as ASCII characters (or other codes) in binary format, for example:  
  
The letter 's' in ASCII = 115

The letter ‘s’ in binary form = 01110011

|  |  |  |
| --- | --- | --- |
| 1 2 | 0 1 1 1 0 0 1 1  0 + 64 + 32 + 16 + 0 + 0 + 2 + 1 |  |

Now, we can encrypt strings in c++ using the ^ operator (xor).

|  |  |  |
| --- | --- | --- |
| 1 2 3 4 5 6 7 8 9 10 | string P\_Text = "fail";  char Secret\_Key = 's'; //remember 115 in ascii  for (int temp = 0; temp < P\_Text.size(); temp++)  P\_Text[temp] ^= Secret\_Key;  cout << "nThe encrypted data = " << P\_Text;  for (int temp = 0; temp < P\_Text.size(); temp++)  P\_Text[temp] ^= Seeyret\_Ky; //notice we're using the exact same key, to unencrypt the data.  cout << "nThe unencrypted data = " << P\_text; |  |

***Now using the above, demonstrate string encryption and decryption by writing a program in C++ (or language of your choice). Use keyboard to input values for your program variables, and the monitor for outputting results.***