

**COMSC 260**  
**Programming Assignment 2**  
**Fall 2020**  
**Worth 15 points (1.5% of your grade)**  
**DUE: Tuesday, 9/15/20 by 11:59 P.M. on**  
**Canvas**

**Start** by downloading the **260\_assign2.cpp** file from the Programming Assignment 2 folder on Canvas

**NOTE:** Your submission for this assignment should be a single **.cpp** file and a single **.pdf** file. The following naming convention should be used for naming your files: **firstname\_lastname\_260\_assign2.cpp** and **firstname\_lastname\_260\_assign2.pdf**. The pdf file that you submit should contain the screenshots of your sample runs of the program (see below). For example, if your first name is “James” and your last name is “Smith”, then your files should be named **James\_Smith\_260\_assign2.cpp** **James\_Smith\_260\_assign2.pdf**.

**COMMENTS (worth 7.5% of your programming assignment grade):** Your program should have at least **ten (10)** different detailed comments explaining the different parts of your program. Each individual comment should be, at a minimum, a short sentence explaining a particular part of your code. You should make each comment as detailed as necessary to fully explain your code. You should also number each of your comments (i.e., comment 1, comment 2, etc.). **Important Note: Any of my comments in the program that you download from Canvas do NOT count towards the ten comments!**

**SAMPLE RUNS (worth 7.5% of your programming assignment grade):** You should submit screenshots of at least **five (5)** different sample runs of your program. Each sample run needs to use different inputs, and your sample runs should **NOT** be the same as the sample runs that are used in this write-up for the assignment. You should also number each of your sample runs (i.e., sample run 1, sample run 2, etc.). All of your sample runs should follow this format – for each individual sample run, screenshot the code with your function calls in main **AND** the corresponding output on the console screen:

```
int main()
{
    cout<<"10000011 binary = "<<bin_to_dec("10000011")<<" decimal\n"; // you should get 131
    cout<<"01010101 binary = "<<bin_to_dec("01010101")<<" decimal\n"; // you should get 85
    cout<<"1111111111111111 binary = "<<bin_to_dec("1111111111111111")<<" decimal\n"; // you
    cout<<"0111111111111111 binary = "<<bin_to_dec("0111111111111111")<<" decimal\n\n"; // y

    cout<<"255 decimal = "<<dec_to_bin(255)<<" binary\n"; //you should get 11111111
    cout<<"65532 decimal = "<<dec_to_bin(65532)<<" binary\n"; //you should get 1111111111111111
    cout<<"12 decimal = "<<dec_to_bin(12)<<" binary\n"; //you should get 1100
    cout<<"1000000 decimal = "<<dec_to_bin(1000000)<<" binary\n\n"; //you should get 11110100

    cout<<"ABC hexadecimal = "<<hex_to_dec("ABC")<<" decimal\n"; //you should get 2,748
    cout<<"F5 hexadecimal = "<<hex_to_dec("F5")<<" decimal\n"; //you should get 245
    cout<<"1234 hexadecimal = "<<hex_to_dec("1234")<<" decimal\n"; //you should get 4,660
    cout<<"FDECB hexadecimal = "<<hex_to_dec("FDECB")<<" decimal\n\n"; //you should get 1,04

    cout<<"512 decimal = "<<dec_to_hex(512)<<" hexadecimal\n"; //you should get 200
    cout<<"5000 decimal = "<<dec_to_hex(5000)<<" hexadecimal\n"; //you should get 1388
    cout<<"900000 decimal = "<<dec_to_hex(900000)<<" hexadecimal\n"; //you should get DBBA0
    cout<<"65525 decimal = "<<dec_to_hex(65525)<<" hexadecimal\n\n"; //you should get FFF5
}
```

```
10000011 binary = 131 decimal
01010101 binary = 85 decimal
1111111111111111 binary = 65535 decimal
0111111111111111 binary = 32767 decimal

255 decimal = 11111111 binary
65532 decimal = 1111111111111100 binary
12 decimal = 1100 binary
1000000 decimal = 11110100001001000000 binary

ABC hexadecimal = 2748 decimal
F5 hexadecimal = 245 decimal
1234 hexadecimal = 4660 decimal
FDECB hexadecimal = 1040075 decimal

512 decimal = 200 hexadecimal
5000 decimal = 1388 hexadecimal
900000 decimal = DBBA0 hexadecimal
65525 decimal = FFF5 hexadecimal

Press any key to continue . . .
```

**NOTE:** The **ONLY** files that should be #included for this assignment are **iostream**, **cmath (for the pow function)**, and **string**. All three of these files are already #included for you in the 260\_assign2.cpp file that you will be downloading from Canvas. **No** other files should be #included for this assignment.

```

// NOTE: The ONLY files that should be #included for this assignment are iostream,
// cmath (for the pow function),
// and string
// No other files should be #included

#include <iostream>
#include <cmath>
#include <string>

// NOTE: The ONLY files that should be #included for this assignment are iostream,
// cmath (for the pow function),
// and string
// No other files should be #included

```

For your second programming assignment you will be writing a **C++ program** with the following functions:

<code>int bin_to_dec(string s)</code>
<b>INPUT(S):</b> an <b>UN</b> signed binary number as a string (e.g., "10110001")
<b>RETURN(S):</b> the binary number converted to the equivalent decimal number
<b>PURPOSE:</b> Converts any <b>UN</b> signed binary number to decimal

<code>string dec_to_bin( int n )</code>
<b>INPUT(s):</b> A non-negative decimal number (e.g, 15)
<b>RETURN(S):</b> The decimal number converted to the equivalent binary number (as a string)
<b>PURPOSE:</b> Converts any non-negative decimal number to binary

<code>int hex_to_dec(string s)</code>
<b>INPUT(S):</b> an <b>UN</b> signed hexadecimal number as a string (e.g., "FABC")
<b>RETURN(S):</b> the hexadecimal number converted to the equivalent decimal number
<b>PURPOSE:</b> Converts any <b>UN</b> signed hexadecimal number to decimal

<code>string dec_to_hex( int n )</code>
<b>INPUT(s):</b> A non-negative decimal number (e.g, 15)
<b>RETURN(S):</b> The decimal number converted to the equivalent hexadecimal number (as a string)
<b>PURPOSE:</b> Converts any non-negative decimal number to hexadecimal

**NOTE:** A major component of this assignment is being able to convert between type char and type int, since binary/hexadecimal numbers are stored as strings, whereas decimal numbers are stored as ints.

For example, we know from the ASCII code table that the character 0, represented as '0', has ASCII code 48. The character 'A' has ASCII code 65, and so on. **You have to handle converting 0 – 9 differently from converting A – F.**

Here are some examples to show the point:

**Example 1: convert any character '0' – '9' to the corresponding number 0 - 9**

```
char char1 = '0'; // the ASCII code for '0' is 48
```

```
int num1 = char1 - 48; // the character '0' is converted to the number 0
```

**Example 2: convert any number 0 – 9 to the corresponding character '0' – '9'**

```
int num1 = 0;
```

```
char char1 = num1 + 48; // the number 0 is converted to the character '0'
```

**Example 3: convert any character 'A' – 'F' to the corresponding number 10 - 15**

```
char char1 = 'A'; // the ASCII code for 'A' is 65
```

```
int num1 = char1 - 55; // the character 'A' is converted to the number 10
```

**Example 4: convert any number 10 – 15 to the corresponding character 'A' – 'F'**

```
int num1 = 10; char char1 = num1 + 55;
```

```
// the number 10 is converted to the character 'A'
```



Link to the **ASCII code table** (bookmark or memorize this link):  
<http://www.asciitable.com/>

### Sample Run:

Here is a sample run of the program. After you finish implementing all four functions, run the program with these test cases (which I've provided to you in the 260\_assign2.cpp file) and make sure you get the same results. The test cases also show you how to properly call each function. As mentioned previously, you will also be coming up with and running five (5) of your own test cases.

```
int main()
{
    cout<<"10000011 binary = "<<bin_to_dec("10000011")<<" decimal\n"; // you should get 131
    cout<<"01010101 binary = "<<bin_to_dec("01010101")<<" decimal\n"; // you should get 85
    cout<<"1111111111111111 binary = "<<bin_to_dec("1111111111111111")<<" decimal\n"; // you
    cout<<"0111111111111111 binary = "<<bin_to_dec("0111111111111111")<<" decimal\n\n"; // y

    cout<<"255 decimal = "<<dec_to_bin(255)<<" binary\n"; //you should get 11111111
    cout<<"65532 decimal = "<<dec_to_bin(65532)<<" binary\n"; //you should get 1111111111111
    cout<<"12 decimal = "<<dec_to_bin(12)<<" binary\n"; //you should get 1100
    cout<<"1000000 decimal = "<<dec_to_bin(1000000)<<" binary\n\n"; //you should get 1111010

    cout<<"ABC hexadecimal = "<<hex_to_dec("ABC")<<" decimal\n"; //you should get 2,748
    cout<<"F5 hexadecimal = "<<hex_to_dec("F5")<<" decimal\n"; //you should get 245
    cout<<"1234 hexadecimal = "<<hex_to_dec("1234")<<" decimal\n"; //you should get 4,660
    cout<<"FDECB hexadecimal = "<<hex_to_dec("FDECB")<<" decimal\n\n"; //you should get 1,04

    cout<<"512 decimal = "<<dec_to_hex(512)<<" hexadecimal\n"; //you should get 200
    cout<<"5000 decimal = "<<dec_to_hex(5000)<<" hexadecimal\n"; //you should get 1388
    cout<<"900000 decimal = "<<dec_to_hex(900000)<<" hexadecimal\n"; //you should get DBBA0
    cout<<"65525 decimal = "<<dec_to_hex(65525)<<" hexadecimal\n\n"; //you should get FFF5
}
```

```
10000011 binary = 131 decimal
01010101 binary = 85 decimal
1111111111111111 binary = 65535 decimal
0111111111111111 binary = 32767 decimal

255 decimal = 11111111 binary
65532 decimal = 1111111111111100 binary
12 decimal = 1100 binary
1000000 decimal = 11110100001001000000 binary

ABC hexadecimal = 2748 decimal
F5 hexadecimal = 245 decimal
1234 hexadecimal = 4660 decimal
FDECB hexadecimal = 1040075 decimal

512 decimal = 200 hexadecimal
5000 decimal = 1388 hexadecimal
900000 decimal = DBBA0 hexadecimal
65525 decimal = FFF5 hexadecimal

Press any key to continue . . .
```