Assignment 1
C/C++ Programming I

C1A1 General Information

Assignment 1 consists of THREE (3) exercises:

C1A1E0 C1A1E1 C1A1E2

All requirements are in this document.

Related examples are in a separate file.

Get a Consolidated Assignment 1 Report (optional)

If you would like to receive a consolidated report containing the results of the most recent version of each exercise submitted for this assignment:

`Send an empty-body email to the assignment checker with the subject line **C1A1_167109_U09609277** and no attachments.

Inspect the report carefully since it is what I will be grading. You may resubmit exercises and report requests as many times as you wish before the assignment deadline.

C1A1 General Information, continued

Course Assignment/Exercise Notation Conventions: Each weekly "assignment" consists of several "exercises". Throughout this course I commonly refer to these using an abbreviated notation, where a form like **C1A2E3** would refer to exercise 3 in assignment 2 of the "C/C++ Programming I" course and **C1A2** would refer to the entirety of assignment 2 of that course.

Getting Started: Before starting your first assignment you must have the appropriate tools for developing your software and the best way to get them is to download and install one of the many free integrated development environment (IDE) packages. These integrate the compiler, editor, and other necessary tools into a convenient GUI application. Although you are free to use any tools you wish and any operating system that will support them, I recommend Microsoft's "Visual Studio Community" for Windows, "Xcode" for macOS, and "Code::Blocks" for Linux. Information on obtaining, installing, and using them is available in the appropriate version of the "Using the … IDE" course document, a link to which is located on the "Assignments" page of the course website. I am sorry but I do not have information on other IDE's or operating systems.

Source Code Files: Header Files and Implementation Files: "Source code" files contain the code necessary for a program to be built without errors and are divided into the two categories "header" files (`.h`, etc.) and "implementation" files (`.c`, `.cpp`, etc.). Not all programs require header files but at least one implementation file is always required. Header files are designed to be included in other files using the `#include` directive but implementation files are not. By placing items that might be needed by multiple other files in header files and including them in those files the bad practice of literally duplicating the needed items in each file can be avoided. Because of their multiple usages, however, header files must never contain anything that will result in an error if more than one file includes them. Files containing data that your program reads or writes are not considered source code files but are instead "data files".

Although some of the following terminology has not yet been discussed in this course it is placed here for completeness and for future reference: Header files typically contain things like macro definitions, inline function definitions, function prototypes, referencing declarations, typedefs, class/structure/union descriptions, and templates, although any of these that will only ever be needed by one specific implementation file may be placed in that file instead. Header files must not contain non-inline function definitions or defining variable declarations; these must be placed in implementation files instead. The header files that are supplied with a compiler provide the information it needs to properly compile code that uses the various functions, macros, and data types available in the compiler's libraries.

Exercise Submission Procedure: Get an exercise to work first on your computer, then submit it to the "assignment checker" and wait for the results to be returned. If there are any errors or warnings make the appropriate corrections and resubmit, repeating as necessary until all issues are corrected. Additional details are provided in each exercise and the course document titled "How to Prepare and Submit Assignments".

C1A1E0 *(6 points total - 1 point per question – No program required)*

Assume language standards compliance and any necessary standard library support unless stated otherwise. These are not trick questions and there is only one correct answer, but basing an answer on runtime results is risky. Place your answers in a plain text "quiz file" named **C1A1E0_Quiz.txt** formatted as:

a "Non-Code" Title Block, an empty line, then the answers:
1. A
2. C
etc.

1. What data type(s) are acceptable for x in the expression *scanf("%e", &x)*
(Note 1.13)
   A. **int** only
   B. **float** only
   C. **double** only
   D. **float**, **double**, and **long double**
   E. any floating point type

2. Which of the following is a string literal?
(Note 1.5)
   A. 'A'
   B. "\"
   C. "\\"
   D. '\x5'
   E. Both "\" and "\\" are string literals.

3. Which of the following is not a legal identifier?
(Note 1.4)
   A. _
   B. **_0**
   C. **T0a1b2c3d4e5555999**
   D. **X**
   E. ***9ABC***

4. Which of the following does not output a numeric value of 50?
(Note 1.12)
   A. **int** x = 50; cout << x;
   B. **char** x = 50; cout << x;
   C. **long double** x = 50; cout << x;
   D. **short** x = 50; cout << x;
   E. *implementation dependent*

5. Which is the most appropriate header file to include in a C++ program to support I/O (input/output) operations such as those performed by *cin* and *cout*?
(Notes 1.9 and 1.10A)
   A. stdio.h
   B. cstdio
   C. cstdlib
   D. iostream
   E. iostream.h

6. If the user types a space followed by ***945*** then presses *Enter*, what value ends up in variable *ch* if the ASCII character set is being used?
   char *ch;*
   scanf("%c", &ch); *or* cin >> ch;
(Notes 1.13, 1.14, and B.1)
   A. 32 if *scanf* or 57 if *cin*
   B. 9 if *scanf* or 945 if *cin*
   C. 945
   D. 9
   E. none of the above

Submitting your solution

`Send an empty-body email to the assignment checker with the subject line **C1A1E0_167109_U09609277** and with your quiz file attached.

See the course document titled "How to Prepare and Submit Assignments" for additional exercise formatting, submission, and assignment checker requirements.

C1A1E1 *(7 points – C++ Program)*

Exclude any existing source code files that may already be in your IDE project and add a new one, naming it **C1A1E1_main.cpp**. Write a program in that file to display two input values in octal, hexadecimal, and decimal. Your program must:

1. not use `cout` more than twice.
2. not use `cin` more than once.
3. not use more than two variables.
4. not use a looping statement.
5. not test anything.
6. use `cout` to prompt (ask) the user to enter two space-separated positive integer numeric values. The first must be in hexadecimal while the second must be in octal. Do not precede the values with 0x or 0.
7. use `cin` to obtain and store and store those values in two type **int** variables.
8. use `cout` to display the first input value in hexadecimal, octal, and decimal in the format shown on the first line below, and the same `cout` to display the second input value in octal, hexadecimal, and decimal in the format shown on the second line below. **D** represents the decimal value, **O** represents the octal value, and **H** represents the hexadecimal value:

```
H hexadecimal = O octal = D decimal
O octal = H hexadecimal = D decimal
```

For example, when run on a machine having a 32-bit type **int**:

a user input of **7 7** produced:

```
7 hexadecimal = 7 octal = 7 decimal
7 octal = 7 hexadecimal = 7 decimal
```

a user input of **1A 17** produced:

```
1a hexadecimal = 32 octal = 26 decimal
17 octal = f hexadecimal = 15 decimal
```

a user input of **abc 777** produced:

```
abc hexadecimal = 5274 octal = 2748 decimal
777 octal = 1ff hexadecimal = 511 decimal
```

Note that `cout` and `printf` both display integer octal and hexadecimal values as unsigned, which is the reason you will get "strange" results if you enter a negative value.

Manually re-run your program several times, testing with at least the following 3 input value pairs:

a 7 **100 100** **def 500**

Submitting your solution

`Send an empty-body email to the assignment checker with the subject line **C1A1E1_167109_U09609277** and with your source code file attached.

See the course document titled "How to Prepare and Submit Assignments" for additional exercise formatting, submission, and assignment checker requirements.

Hints:

See notes 1.12 and 1.14. Use the `hex`, `oct`, and `dec` manipulators to change the integer input and output format to hexadecimal, octal, or decimal, respectively. The selected format will remain in effect until explicitly changed. Upon program startup the format is always decimal.

C1A1E2 *(7 points – C Program)*

Exclude any existing source code files that may already be in your IDE project and add a new one, naming it **C1A1E2_main.c**. Write a program in that file that prompts (asks) the user to enter any decimal value then evaluates the following expression with that user value in **x**:

$3x^3 - 5x^2 + 6$

Display the results in the following format, where **U** represents the user input value and **R** represents the value of the expression:

```
If x = U the value of "3x^3 - 5x^2 + 6" is R
```

Specifically, if the user inputs a value of `-2.2` the exact display would be:

```
If x = -2.2 the value of "3x^3 - 5x^2 + 6" is -50.144
```

Note that the ^ symbol is being used to indicate exponentiation since it is not always possible to produce a superscript in a plain text display. While the C/C++ ^ operator does not actually perform exponentiation, it does in some languages and is sometimes used in computer literature to indicate it.

Your program must:

1. use type `double` for all variables.
2. use `printf`'s `%g` conversion specifier to display the input and resultant values.
3. not call any functions other than `printf` and `scanf`.
4. not call `printf` more than twice.
5. not call `scanf` more than once.

Submitting your solution

`Send an empty-body email to the assignment checker with the subject line **C1A1E2_167109_U09609277** and with your source code file attached.

See the course document titled "How to Prepare and Submit Assignments" for additional exercise formatting, submission, and assignment checker requirements.

Hints:

To represent a double-quote in any string literal use a backslash followed by a double-quote. The compiler automatically concatenates multiple string literals separated only by zero or more whitespaces into one string, including string literals on separate lines.

If your rusty on your basic 8th/9th grade "Algebra I" and do not fully understand what $3x^3 - 5x^2 + 6$ means, please contact me with your question(s).

