

Honors Computer Science C++

Mr. Clausen

Program 16A, 16B, 16C

Special Note: Every program from Chapter 4 to the end of the year needs to have functions!

Program 16A MadLibs (30 points)

Write a program that asks the user to enter 2 different nouns, verbs, adverbs, and adjectives.

Make your program user friendly by prompting them for these values. Use functions and one line comments to separate this program into its parts: Get_Data (input), calculations (concatenating the sentences) and output (a function to display the sentences)

(Parts of Speech Review: http://www.englishclub.com/grammar/parts-of-speech_1.htm)

1. Type comments at the beginning of the program to display your name and other information just like those used for program 4A. **Make sure to change the program name and program description in these comments, so that the program number, name, and description say what is listed above.**
2. Include `<iostream>` so we can use the **cout** and **cin** commands.
3. **Type:** using namespace std;
4. There are no constants to declare.
5. Don't forget to include comments for your functions, right before each of your function declarations.
6. Declare all of the functions necessary for this program including `Display_My_Info ()`, `Get_Data()`, a calculations function (concatenating string literals and the parts of speech variables) and `Display_Output()`.
7. Use a comment line with **equal signs** to separate all of the above from the `int main` function. For example:

```
//=====
```
8. Inside the **int main()** function, declare **and initialize** all of your variables and then call the functions. **Don't forget that value-returning functions are "called" in assignment statements, while void functions can be called just by writing the function name including any of the "actual" parameters.**

Don't forget your return 0 command.

9. After the **int** main() function, have another comment line of **subtraction signs** to separate the above from your function implementation lines below. For example:
//-----
10. Implement all of the functions, separating each one from the other ones using comment lines of **subtraction signs**.
11. In the Get_Data() function, ask the user to enter noun1, verb1, adverb1, adjective1, etc.
12. For the “calculations function”, perform all of the string processing “calculations” while assigning the results to variables. The variable sentence1 should be assigned string literals (your preselected words for the Madlib in quotes) interspersed between the variables noun1, verb1, adverb1, and adjective1 using concatenation. Continue this pattern for sentence2.
13. For the Display_Output() function, display the sentences.
14. After the last function implementation of every program, end your program with two comment lines of equal signs. This signifies the end of your source code. For Example:
//=====

When you are finished with your program, have tested it thoroughly to make sure that everything is correct, and are sure that you don't need to make any changes, then save your program in the “T” network mapping, and the Program 16A folder.

Program 16B Be My Pal-indrome (25 points)

This program should ask the user to enter **one lower case** word and test it to see if it is a palindrome. **Include a loop in the main function so we can test more than one word (see step #8)**. A palindrome is a word that is the same forwards and backwards. For example: mom, dad, racecar, etc.

1. Type comments at the beginning of the program to display your name and other information just like those used for program 4A. **Make sure to change the program name and program description in these comments, so that the program number, name, and description say what is listed above.**

2. Include `<iostream>` so we can use the **cout** and **cin** commands.
3. **Type:** using namespace std;
4. There are no constants to declare.
5. Don't forget to include comments for your functions, right before each of your function declarations.
6. Declare all of the functions necessary for this program including `Display_My_Info()`, `Get_Data()`, `Display_Output()`, and any other functions that you may need. Remember that functions should only do one task.
7. Use a comment line with **equal signs** to separate all of the above from the `int main` function. For example:

```
//=====
```
8. Inside the `int main()` function, declare **and initialize** all of your variables and then call the functions.
In the main function include a while loop with 2 menu choices:
1) Palindrome Test, or Q) quit. If you wish, you may add a "menu function" to display these choices.
 Don't forget that value-returning functions are "called" in assignment statements, while void functions can be called just by writing the function name including any of the "actual" parameters.
Don't forget your return 0 command.
9. After the `int main()` function, have another comment line of **subtraction signs** to separate the above from your function implementation lines below. For example:

```
//-----
```
10. Implement all of the functions, separating each one from the other ones using comment lines of **subtraction signs**.
11. In the `Get_Data()` function, ask the user to enter one word and "return" it to the main function.
12. For the "calculations function", perform all of the string processing "calculations". These calculations are going to be a little different than just crunching numbers. There is more than one algorithm to determine whether the word is a palindrome or not. Feel free to choose whichever algorithm you wish. Since there shouldn't be any "cout" statements in this function, I suggest using a Boolean variable which is assigned a **true** or **false** depending whether the word is a palindrome or not.

13. For the Display_Output() function, echo (cout) the word, then tell the user whether the word is a palindrome or not.
14. After the last function implementation of every program, end your program with two comment lines of equal signs. This signifies the end of your source code. For Example:

```
//=====
//=====
```

When you are finished with your program, have tested it thoroughly to make sure that everything is correct, and are sure that you don't need to make any changes, then save your program in the "T" network mapping, and the Program 16B folder.

Program 16C Pig Latin Translator (30 points)

This program should ask the user to enter **one lower case** word and translate it to Pig Latin. **Include a loop in the main function so we can test more than one word (see step #8).** Here are the rules for translating a word from English into Pig Latin. For words that begin with vowels (a, e, i, o, u, don't worry about y), just add "ay" to the end of the word (we are going to do a simple translation, so we won't add "yay" or "way" to the end). For example, "apple" becomes "appleay" in Pig Latin. For words that begin with a consonant or consonant clusters (multiple consonants that form one sound), all letters before the first vowel are removed from before the first vowel and placed at the end of the word, then "ay" is added to the end of that. For example, "smile" becomes "ilesmay", and "computer" becomes "omputercay" in Pig Latin. For words that don't have vowels like the word "sky", just add an "ay" at the end of the word. This is not perfect Pig Latin, but a good simplification.

1. Type comments at the beginning of the program to display your name and other information just like those used for program 4A. **Make sure to change the program name and program description in these comments, so that the program number, name, and description say what is listed above.**
2. Include <iostream> so we can use the **cout** and **cin** commands.
3. **Type:** using namespace std;
4. There are no constants to declare (unless you wish to make a constant for "ay").
5. Don't forget to include comments for your functions, right before each of your function declarations.

6. Declare all of the functions necessary for this program including `Display_My_Info()`, `Get_Data()`, `Display_Output()`, and any other functions that you may need. Remember that functions should only do one task.
7. Use a comment line with **equal signs** to separate all of the above from the `int main` function. For example:

```
//=====
```
8. Inside the `int main()` function, declare **and initialize** all of your variables and then call the functions.
In the main function include a while loop with 2 menu choices:
1) Convert a word to pig latin, or Q) quit. If you wish, you may add a “menu function” to display these choices.
 Don't forget that value-returning functions are “called” in assignment statements, while void functions can be called just by writing the function name including any of the “actual” parameters.
Don't forget your return 0 command.
9. After the `int main()` function, have another comment line of **subtraction signs** to separate the above from your function implementation lines below. For example:

```
//-----
```
10. Implement all of the functions, separating each one from the other ones using comment lines of **subtraction signs**.
11. In the `Get_Data()` function, ask the user to enter one word and “return” it to the main function.
12. For the “calculations function”, perform all of the string processing “calculations”. Send the word to be translated into this function, and return the translated Pig Latin word from this function. There is more than one algorithm to convert a word to Pig Latin. Feel free to choose whichever algorithm you wish. There shouldn't be any “cout” statements in this function, so return the Pig Latin word to the main function.
13. For the `Display_Output()` function, echo (cout) the original word, then tell the user the equivalent word in Pig Latin.
14. After the last function implementation of every program, end your program with two comment lines of equal signs. This signifies the end of your source code. For Example:

```
//=====
//=====
```

When you are finished with your program, have tested it thoroughly to make sure that everything is correct, and are sure that you don't need to make any changes, then save your program in the "T" network mapping, and the Program 16C folder.