CSC 150 Program 1
Assigned: 9/21/2007 Due: 10/19/2007
This version is here for convenience.
The official document can be found at the Campus CSC 150 Page

Description:

Living in the digital age, people have come to expect computers to do many things for us. We have online books, e-mail, instant messaging, and much more. But many of these luxuries have been unavailable to people with sight problems. To accommodate them, many companies have developed special software to take digital text and convert it to sound so the seeing impaired can listen to what we see every day. This is typically referred to as text to speech.

Consider the problem of a sight impaired person making or getting change from a purchase. Cash registers will display the amount to be given back, but it's up to the person to figure out how many dollars and how many of each coin type is needed. Your program will help in this respect, and can also be used as a training aid for cashiers.

This program will ask the user to enter the amount of the purchase, and then the amount paid. If the payment is not enough, ask the user to enter additional payment. If this combined amount is still insufficient, give an error message and exit the program. If the payment amount is enough, calculate the change due and "speak" the amount of dollars and of each coin due.

For this assignment, assume the largest amount of change due will not exceed $9.99.

Getting Started:

Save the file "prog1_files.exe" to the folder where you want the prog1 project stored. Double-click the file to start the extraction process, it will run automatically. This will create the project folder, with a file prog1.cpp as the file you will write your program in. You can start working on the project by double clicking the prog1.sln file. Do not delete any of the files that are created for you, and do not modify any part of prog1.cpp other than the function main( ) and your program header information

The text to speech code is contained in the files sapi.h and sapi.lib. You don't need to be concerned with their content or how they work. The function speech( ) in your program takes care of the details.

Using the speech function:

Warning, do not attempt to change the function speech( ) that is provided at the bottom of your code. Doing so will probably result in not being able to compile and run your code correctly.

You can pass any legal word to our function and it will be output to the speakers and to the screen for you. Sounds tough, not really. All you need to understand is how to call the function that is already coded for you.

To output a single word to the speakers, just insert the following code in your program and it will be piped to the speakers.

```
    speech("hello ");
```

If you wish a phrase of words to be piped to the speakers, you could call the speech function with the following code.

```
    speech("Hello class ");
```

So when you see a digit like 9, you could call the speech function with the following code to get nine to be piped to the speakers.

```
    speech("nine ");
```

If the number was 70, you could call the speech function with the following code to get seventy to come to the screen.

```
    speech("seventy ");
```

NOTE: Our speech program actually pipes the word to the speakers and to the console screen. The speech function does this so if your PC does not have speakers, you can see what would have been output. In the function calls above, you will note the space after the word/phrase. This is done to provide separation of words that come to the screen. You must enclose the text you want spoken inside double quote marks ("like this").

The General Algorithm:

1. Read in floating point purchase amount from the user
2. Read in a floating point payment amount from the user
3. If payment is not enough, ask for more
4. If original plus additional payment is not enough, give message and exit program
5. Compute change due − convert values to integers
6. Determine number of dollars, quarters, dimes, nickels, pennies due (use division and modulus)
7. Output the number of each money unit due
8. End with a thank you message.

You will find that once you have the code to speak one of the money units, you will use a copy of that with modification for the name of units for each of the others. At this point in the course, this copy and paste method is what you must use, although, as we'll discuss later on, this duplication can easily be avoided when you learn how to write your own functions.

Sample run:

```
Welcome to Tech Audible Change Maker
Enter purchase amount 12.71
Enter payment amount 15
two dollars
one quarter
no dimes
no nickels and
four pennies
Thank you.
```

Note that when the number of some coin to be returned is zero, use the word "no" or "zero", do not skip that coin. You can embed the escape code '\n' in the text you send to the speech( ) function in order to get the items on separate lines. This code will not be spoken. You should use correct singular and plural versions of the dollar and coin names.

Timeline:

Fri, 9/28 − give prompts, get inputs, check payment.

Fri, 10/5 − calculate amount of each coin/dollars due.

Week of 10/1-10/5 − PROGRESS CHECK. You must bring in a print out of your program completed so far to your lecture instructor. You must hand it in IN PERSON. Do so when you will have sufficient time to discuss your progress and resolve any problems.

Fri, 10/12 − be able to speak each of the values required.

Fri, 10/19 − the complete program is due. Your code should be working and be fully commented and indented according to the coding standards.

Comments and Hints:

1. Work on it in small sections.
2. If you run into problems and can not find an error, seek help. The lecture instructors and lab TA's are available during the weekdays with a wide range of office hours.
3. Try many numbers on your program. Don't assume that once you get one section working that when you make a change, it will still work. This often is not the case.
4. If you don't have speakers on your computer, don't worry, the words will be displayed on the screen.
5. When seeking help, always have a digital copy of your code available. A printout is also valuable.
6. Save your work often.
7. Be sure to follow the coding style guidelines that can be found on the file server at F:\dept\mcs\csc150\course info\coding_style.doc.
8. Be sure your code file is readable and neat. Do not allow lines to extend past 80 characters, use appropriate white space and make sure to use a consistent and attractive indentation scheme.
9. Your program must correctly compile in the environment in the labs (Visual C++ 2005 running in Windows XP).

Program Submission

Submit your program code (the prog1.cpp file only) at www.mcs.sdsmt.edu/view.php?p=2600 before midnight of the due date. (Your file gets time stamped, so late submissions will be noted and will not be graded!) Be sure to submit to the correct lecture section!

Note that when using the Submit It page, completely replace the labels in the name boxes "first name" and "last name" with your own name. Choose the correct instructor, press the Select Instructor button, then choose the appropriate lecture section. Submit only your prog1.cpp file

Timestamp 2/7/2017

Website: http://www.mcs.sdsmt.edu/ecorwin/cs1/fall07/program1.html