DeVry University

Student Lab Activity

A. Lab # CIS CIS170C-A4

B. Lab 4 of 7: Functions

C. Lab Overview – Scenario/Summary

You will code, build, and execute a program that simulates the dialing of a phone using functions.

Learning outcomes:

1. Distinguish between pass by value and by reference.
2. Call functions using &.
3. Write functions using value and reference.
4. Be able to define and use global named constants.
5. Be able to debug a program with syntax and logic errors.
6. Be able to use the debug step-into feature to step through the logic of the program and to see how the variables change values.

D. Deliverables

Section	Deliverable	Points
Lab 4	Step 5: Program Listing and Output	**45**

E. Lab Steps

Preparation:
If you are using the Citrix remote lab, follow the login instructions located in the iLab tab in Course Home.

Locate the Visual Studio 2010 icon and launch the application.

Lab:

Step 1: Requirements: **Phone-Dialing Program**
Write a program that simulates the dialing of a phone number. A user will input an 8-place number, for example: 359-3177 (note that the hyphen is considered a digit). The rules for entering phone numbers follow. • 8 places

- It may have numbers, letters, or both.
- The phone number cannot begin with 555.
- The phone number cannot begin with 0.
- The hyphen must be in the 4th position.
- No other characters (@#$%^&*()_+=\|/><etc.) are allowed.
- If a letter is entered, its output will be a number (check your phone pad).
- Enter Q to Quit.

If all of the rules are met, you will output a message to the console that reads like the following.
**Phone Number Dialed: UN9-3177 ***the number entered

If all of the rules are not met, then you output one of the following error messages to the console.

- **ERROR - Phone number cannot begin with 555**
- **ERROR - Phone number cannot begin with 0**
- **ERROR - Hyphen is not in the correct position**
- **ERROR - An invalid character was entered**

It will then prompt the user to try again.

This should be a lot of fun!
Here are some great things to think about as you begin your program!

Define a function named **ReadDials()** that reads each digit and letter dialed into 8 separate char variables (DO NOT USE ARRAYS). All the digits are sent back **through parameters by reference**.

Then, for each digit, the program will use a function named **ToDigit()**, which receives a single char argument (pass by reference) that may be a number or a letter of one of the digits dialed.

If it is a number, then return 0 by value indicating that it is a valid digit. If the digit is a letter, then the number corresponding to the letter is returned by reference, and return 0 by value indicating that it is a valid digit. Here are the letters associated with each digit.

0		5	J K L
1		6	M N O
2	A B C	7	P Q R S
3	D E F	8	T U V
4	G H I	9	W X Y Z

If the digit entered is not one of the valid digits or one of the valid letters, return –1 by value indicating that you have an invalid digit.

A phone number never begins with a 0, so the program should flag an error if such a number is entered. **Make ReadDials() return –2 in this case**.

A phone number never begins with 555, so the program should flag an error if such a number is entered. **Make ReadDials() return –3 in this case**.

A phone number always has a hyphen (-) in the 4th position. **Make ReadDials() return –4 in this case** (if it doesn't have a hyphen in the 4th position). If a hyphen is in any other position, it is considered an invalid digit.

If the phone number is valid, the main calls the **AcknowledgeCall function** to write the converted number to the output file.

All the logic of the program should be put in functions that are called from Main(): **ReadDials()** and **AcknowledgeCall()**.

The **ToDigits()** function is called from the **ReadDials() function** and is used to convert each letter entered individually into a digit and to verify that the user has entered a valid phone number. Have the program work for any number of phone numbers.

In the ToDigits() function uses the **toupper function** to convert any letters entered to uppercase. All the error messages are to be written to the output file from main() based on the return value from the functions.

Continue processing until the user enters a Q.

You will set up the **8 char variables** to hold the digits of the phone number in main() and pass the variables to the functions by reference.

Sample Output from the Program

```
Enter a phone number (Q to quit): 213-2121
Phone Number Dialed: 213-2121

Enter a phone number (Q to quit): asc-dfer
Phone Number Dialed: 272-3337

Enter a phone number (Q to quit): 555-resw
ERROR - Phone number cannot begin with 555

Enter a phone number (Q to quit): 098-8765
ERROR - Phone number cannot begin with 0

Enter a phone number (Q to quit): 12345678
ERROR - Hyphen is not in the correct position

Enter a phone number (Q to quit): @34-*uyt
ERROR - An invalid character was entered

Enter a phone number (Q to quit): Q
Press any key to continue . . .
```

Step 2: Processing Logic

Using the pseudocode below, write the code that will meet the requirements.

```
Main Function
   Declare the char variables for the 8 digits of the phone number

   while true
      Call the ReadDials function passing the 8 digits
      by reference.  ReadDials returns an error code by
      value.

      If the return value is -5, exit the do while loop

      If the error code is -1, display the
        error message "ERROR - An invalid character was entered".
      If the error code is -2, display the
        error message "ERROR - Phone number cannot begin with 0".
      If the error code is -3, display the
        error message "ERROR - Phone number cannot begin with 555".
      If the error code is -4, display the
        error message "ERROR - Hyphen is not in the correct position".
      Otherwise, call the AcknowledgeCall function
   End-While

ReadDials function

   Input the first digit
   If a Q was entered, return -5.
   Input the rest of the phone number

   Call the ToDigit function for each of the 7 digits
     not for digit 4
   If ToDigit returns -1, return -1
   If digit 4 is not a hyphen, return -4.
   If digit 1 is 0, return -2.
   If digits 1 - 3 are 5, return -3
   Otherwise, return 0

ToDigit function
   Convert the digit to upper case
   Use a switch statement to determine if the digit is valid
     and convert the letters to digits

   If the digit is invalid, return -1.
   If the digit is valid, return 0.

AcknowledgeCall function
   Display the Phone Number.
```

Step 3: Create a New Project

Create a new project and name it LAB4.

Write your code using the processing logic in Step 2 (above). Make sure that you save your program.

Step 4: Compile and Execute

a) Compile your program. Eliminate all the syntax errors.

b) Build your program and verify the results of the program. Make corrections to the program logic, if necessary, until the results of the program execution are what you expect.

Step 5: Print Screen Shots and Program

1. Capture a screen print of your output (do a print screen and paste into an MS Word document).
2. Copy your code and paste it into the same MS Word document that contains the screen print of your output.
3. Save the Word document as Lab04_LastName_FirstInitial.

END OF LAB