Do-While and For Loop Programming Example

**Problem Description:** Meteorologists have recorded monthly rainfall amounts at several sites throughout a region of the country. You have been asked to write a program that lets a user enter a years worth of rainfall amounts at a particular site. The average rainfall for that site is then printed to the screen. After processing one site, the user is prompted to see if another site needs processing. A y for yes and an n for no are to be entered at the prompt.

**Input:** For each recording site, 12 floating-point rainfall amounts are entered by the user (could be modified for reading from a file). User is also prompted if they want to continue with another sites information.

**Output:** The average rainfall for each site is written to the screen.

**Discussion:** Looping structures are used to solve this problem. Some of these loops are used to ensure valid inputs as specified by the program. At the top level, a loop must be created to process the information for each site. The number of sites is not known ahead of time, and we assume at least one site is to be processed. Therefore a do-while loop is used. At the end of this loop, the user is prompted to see if another site needs to be processed.

Another loop is required to process the 12 monthly rainfall values for a particular site. Before the loop, we initialize a sum to 0. Use a for loop for reading these values since we know that 12 iterations are required.

Two other loops are required. One to verify that the rainfall entered is not less than 0 and another to ensure that 'y’ or ‘n’ is entered at the continue prompt. For these loops, do-while loops work very well.

**Assumptions:** Data for at least one site is to be entered. All information entered is to be tested and verified to be valid before being used. In most cases, valid data types and no extraneous information is assumed for values entered. The testing is for verification that the value entered is valid.
Main – Rainfall calculation
Do
  Get 12 rainfall amounts(function) and sum them
  Print out the sum/12.0
  Prompt user to continue
    GetResponse(function)
While response to prompt is ‘y’

Get12Amounts (OUT: float& sum)
  Initialize sum to 0
  For loop count is 1 to count equals 12
    Prompt for a rainfall amount
      GetOneAmount(function)
    Increase sum by amount read

GetOneAmount (OUT: float& amount)
  Do
    Read amount;
    If amount < 0
      Print out less than 0 message, re-enter
    While amount < 0

GetResponse (OUT: char& response)
  Do
    Read response
    Test response for ‘y’ or ‘n’ – if it is not one,
      Output error message and re-prompt
    While response is not ‘y’ or ‘n’
Program Output:

E:\CPE112\Summer06\Lecture_Programs\Ch_09>rainfall.exe
Enter rainfall amount 1: 22
Enter rainfall amount 2: 23
Enter rainfall amount 3: -1
Amount cannot be negative. Enter again: 1
Enter rainfall amount 4: 55
Enter rainfall amount 5: 23
Enter rainfall amount 6: 6
Enter rainfall amount 7: 3
Enter rainfall amount 8: 1
Enter rainfall amount 9: 4
Enter rainfall amount 10: 33
Enter rainfall amount 11: 54
Enter rainfall amount 12: 23

Average rainfall is 20.67 inches

Do you have another recording site? (y or n) K
Please type y or n: Y
Please type y or n: n

E:\CPE112\Summer06\Lecture_Programs\Ch_09>