Program Flow Charting

How to tackle the beginning stage of a program design
A Program

Set of instructions written in a programming language that tells the computer what to do
Programmers

• Prepare instructions that make up the program
• Run the instructions to see if they produce the correct results
• Make corrections
• Document the program
• Interact with
  - Users
  - Managers
  - Systems analysts
• Coordinate with other programmers to build a complete system
The Programming Process

• Defining the problem
• Planning the solution
• Coding the program
• Testing the program
• Documenting the program
The Programming Process: Defining the Problem

• What is the input
• What output do you expect
• How do you get from the input to the output
The Programming Process: Planning the Solution

- Algorithms
  - Detailed solutions to a given problem
    - Sorting records, adding sums of numbers, etc..

- Design tools
  - Flowchart
  - Pseudocode
    - Has logic structure, but no command syntax
The Programming Process: *Planning the Solution*

- Desk-checking
  - Personal code design walk through
- Peer Reviews
  - “Code walk through”/structured walk through
Flow Control Elements

The Programming Process: *Planning the Solution*
The Programming Process: Planning the Solution

Accept series of numbers and display the average
The Programming Process: Coding the Program

• Translate algorithm into a formal programming language
• Within syntax of the language
• How to key in the statements?
  - Text editor
  - Programming environment
    • Interactive Development Environment (IDE)
The Programming Process: Testing the Program

• Translation - compiler
  - Translates from source module into object module
  - Detects syntax errors

• Link - linkage editor (linker)
  - Combines object module with libraries to create load module
  - Finds undefined external references

• Debugging
  - Run using data that tests all statements
  - Logic errors
The Programming Process: Documenting the Program

- Performed throughout the development
- Material generated during each step
  - Problem definitions
  - Program plan
  - Comments within source code
  - Testing procedures
  - Narrative
  - Layouts of input and output
  - Program listing
Procedural Level Languages

• 1\textsuperscript{st} Generation: Machine Level
• 2\textsuperscript{nd} Generation: Assembly Level
• 3\textsuperscript{rd} Generation: High Level
FORTRAN

This program will find the average of integers you enter through the keyboard. Type 999 to indicate end of data.

Please enter a number 6
Please enter the next number 4
Please enter the next number 11
Please enter the next number 999

The average of the numbers is 7.00
COBOL
Third Generation Languages

'BASIC PROGRAM
' AVERAGING INTEGERS ENTERED THROUGH THE KEYBOARD
CLS
PRINT "THIS PROGRAM WILL FIND THE AVERAGE OF INTEGERS YOU ENTER"
PRINT "THROUGH THE KEYBOARD. TYPE 999 TO INDICATE END OF DATA."
PRINT
SUM=0
COUNTER=0
PRINT "PLEASE ENTER A NUMBER"
INPUT NUMBER
DO WHILE NUMBER <> 999
    SUM=SUM+NUMBER
    COUNTER=COUNTER+1
    PRINT "PLEASE ENTER THE NEXT NUMBER"
    INPUT NUMBER
LOOP
AVERAGE=SUM/COUNTER
PRINT "THE AVERAGE OF THE NUMBERS IS"; AVERAGE
END

(a)

(b)

THIS PROGRAM WILL FIND THE AVERAGE OF INTEGERS YOU ENTER THROUGH THE KEYBOARD. TYPE 999 TO INDICATE END OF DATA.

PLEASE ENTER A NUMBER
?6
PLEASE ENTER THE NEXT NUMBER
?4
PLEASE ENTER THE NEXT NUMBER
?11
PLEASE ENTER THE NEXT NUMBER
?999
THE AVERAGE OF THE NUMBERS IS 7
Third Generation Languages

// C++ PROGRAM
// AVERAGING INTEGERS ENTERED THROUGH THE KEYBOARD

#include <iostream.h>

main ()
{
    float average;
    int number, counter = 0; int sum = 0;
    cout << "THIS PROGRAM WILL FIND THE AVERAGE OF INTEGERS YOU ENTER \n";
    cout << "THROUGH THE KEYBOARD. TYPE 999 TO INDICATE END OF DATA. \n";
    cout << "PLEASE ENTER A NUMBER";
    cin >> number;
    while (number != 999)
    {
        sum = sum + number;
        counter ++;
        cout << " \nPLEASE ENTER THE NEXT NUMBER";
        cin >> number;
    }
    average = sum / counter;
    cout << " \nTHE AVERAGE OF THE NUMBERS IS " << average
}

(a)

(b)