Tuesday, July 11, 2017

C++ Primer Plus

Branching Statements and Logical Operators

IF

- If and if else
- Else if

LOGICAL EXPRESSION

- Logical OR ||, logical AND &&, logical NOT !
- Logical OR - ||
  - At least one true
  - || is a sequence point
    - Any value changes indicated on the left side take place before the right side is evaluated
- Logical AND - &&
  - Both of the sides must be true
- The logical NOT - !
- Logical operator facts
  - Logical OR and logical AND have lower precedence than relational operators
  - Logical NOT has higher precedence than relational operators
  - Logical AND has higher precedence than logical OR
  - C++ guarantees evaluation of a logical expression from left to right
  - Logical AND
    - Program doesn't evaluate second condition if first one is false
- Alternative representations
  - Not all keyboards provide all the symbols used for the logical operators
  - And, or, not - C++ reserved words

- They aren't considered keywords because they are alternative representations

THE cctype LIBRARY OF CHARACTER FUNCTIONS

- Character-related functions
  - Determine whether character is uppercase, digit, punctuation, …
- isalpha(ch)
  - Returns a nonzero value if ch is a letter and zero value otherwise
- ispunct(ch)
  - Returns true if ch is a punctuation character
  - Returns int
- It is better to use it because computer might not use ASCII
  - So this might not work on every computer

THE ?: OPERATOR

- Instead of if else statement
- Conditional operator ?:
- If expression1 is true -> value of whole conditional expression is expression2
- Otherwise -> expression3

SWITCH

- Default case is optional
- When program jumps to a particular line, then it executes all remaining lines
  - unless you stop it (break)
- You can use enumerators as labels
- Switch case label must
  - Be single value and integer (char included)
- Switch statement can't handle floating point tests
- Break and continue statements
  - Enable program to skip over parts of the code

- Break
  - In switch, loops
  - Pass to the next statement following the switch or the loop
- Continue
  - Skip rest of the body and start new loop cycle

NUMBER-READING LOOPS

- Number-reading loops
  - What happens when user responds by entering a word instead of a number?
    - Value left unchanged
    - Mismatched input is left in the input queue
    - Error flag is set in the bin object
      - You have to reset the flag before the program can read more input
      - clear() method resets the bad input flag (also resets the end of file condition)
    - Call to the cin method, if converted to type bool, returns false
      - You can use nonnumeric input to terminate a number reading loop
  - What the solution is when type don't match
    - Use the value of a cin input expression to test for non-numeric input
    - Let user try again
      - Reset cin to accept new input
    - Get rid of the bad inout
    - Prompt the user to try again

SIMPLE FILE INPUT/OUTPUT

- Console I/O = display output to file input
- File I/O = display output to file output
- When you use cin for input, program views as a series of bytes
  - Each byte being interpreted as a character code

    - No matter what destination data type is, input begins as character data
- On output opposite translations take place
    - Integers -> converted to sequences of digit characters
    - Floating-point numbers -> sequences of digits and other characters
    - Character data -> no translation
- All input starts out as text
    - File equivalent to console input is a text file
    - But not all files are text files
        - Databases, speadsheets, word processing files
    - Use Notepad for Windows, emacs for Linux,
- Writing to a text file
    - #include <fstream>
        - fstream header file defines ofstream class for handling the output
    - Define at least one ofstream variables, or objects
    - Std namespace
        - Using directive or std:: prefix for elements such as ofstream
    - Associate of stream object with a specific file
        - Use open()
            - Requires C-style string as its argument (literal string or string stored in an array)
    - Use objects
        - After you have declared object and associated it with a file, you use it exactly as you would use cout
            - <<, endl, setf() are also available to ofstream objects
    - When you finish with a file
        - Use close()

    - If you forget, it will be done automatically (if program terminates normally)
    - Doesn't require filename (because outFile was already associated with a particular file)
  - Use an ofstream object with the >> operator to output a variety of data types
- Example
  - It will generate new file carinfo.txt
    - Contains output generated by program
    - Check directory or folder that contains the executable program
- outFile
  - Can use the same methods as cout
- open()
  - If file doesn't exist, open() will create it
  - If file exists, open() truncates the file (discard contents) then put in new contents
- Reading from a text file
  - #include <fstream>
    - Defines ifstream class for handling input
  - Declare one or more ifstream variables, or objects
  - Std namespace
  - Associate ifstream object with a specific file
    - open()
  - Use objects
  - When you are finished with file, you close() it
  - You can use ifstream object with the << operator, get(), getline() to read
    - eof(), fail() - monitors the success of an input attempt
  - Ifstream object itself when used as a test condition is converted to boolean
    - True - if last read attempt succeeded

- What happens if you try to open nonexistent file
- Example
- eof() returns true if end of file was reached