Thursday, June 29, 2017

C++ Primer Plus

Dealing with Data

- Naming rules
  - Letters, numbers, underscore (_)
    - Can't start with number
  - Case sensitive
  - No C++ keywords
    - Main can be used as name of the variable
  - Names beginning with 2 underscore characters or with an underscore character followed with capital letter
    - Reserved for compiler
    - __number, _Number
    - Doesn't produce a compiler error
    - Leads to undefined behavior - non predictable results
  - Names beginning with single underscore character
    - Reserved for global identifiers
    - _number
  - No length limit, all characters are significant
    - C - just first 63 characters are significant
- Integer types
  - Short (at least 16 bits)
  - int (at least as big as short)
    - Most natural type - computer hands it most efficiently
  - long (at least 32 bits wide and at least as big as int)
  - Choose width according to the computer design

  - sizeof operator
- sizeof operator
  - Maximum and minimum value of type
- #include <climits> header
  - It defines INT_MAX, INT_MIN, …
- If you don't define variable
  - Value is undefined
    - The value is whatever was in the memory location prior the creation
- unsigned = unsigned int
- #define ZERO 0
  - Makes ZERO symbol for 0 value
- C99 added new types to the C++ Standard
  - (unsigned) long long - at least 64 bits and at least as big as (unsigned) long
- C++ lets you write integers in 3 number bases
  - 10 (public)
  - 8 (old Unix) - first two digits 0, 032
  - 16 (hardware) - first two ox or oX, ox42
  - Cout display all in decimal form
- Cout - display in octal and hexadecimal formats
  - You can also do std::hex if you omit using namespace std
- Suffixes
  - L or l
    - type long constant
  - U or u
    - Unsigned int constant
  - ul (any order of letters and cases)

    - Unsigned long constant
- Char
    - Char ch = ‘M’
        - cout.put(ch) displays it
    - Integer type - stores characters, letters, digits
    - Is not signed by default, nor unsigned
        - Specify only if you use char to store numbers
    - Most systems support fewer than 256 characters (ASCII, IBM mainframe, EBCDIC)
    - wchar_t
        - Character that needs more than 8 bits (Japanese)
        - Wide character typed
- You type letter M -> cin converts M to 77 -> program stores it as 77 -> cout converts 77 to M -> prints M
- If you want to find out ASCII value for M
    - Int value = m
- Member function
    - Belongs to a class and describes a method for manipulation class data
    - cout.put()
    - You can use member function only with a particular object of that class (cout object)
- Escape sequence
    - Special notations for special characters
    - \n … new line
    - Numeric escape sequence
        - Tied to a particular code (\0x8)
    - Symbolic escape sequence

    - Works with all codes (\b)
- Universal character names
    - C++ implementations support a basic source character set and basic execution character set
        - Set of character you can write code with and characters can be produced by the execution of a program
    - Using universal character names is similar to using escape sequences
        - Your implementation has to support it
- Unicode
    - ASCII is subset of Unicode
    - More than 96000 symbols, 49 scripts
- Boolean type
    - In the past C and C++ didn't have boolean
    - bool test = true;
- Const quantifier
    - const int HOURS = 24;
    - Provide a value when you declare constant
        - Otherwise it ends up with an specifics value that you can't modify
    - You can also use #define
        - Scope difference
    - In C++, not C, you can use a const value to declare the size of an array
- Floating point numbers
    - Greater range in values
    - You can represent numbers like that 3.24e3 = 3.24*1000
        - 3.24 - mantissa, 3 - exponent
        - Can't have spaces (3e 2)

- Types
  - Float - at least 32 bits
  - double - at least 48 bits and no smaller than float
  - long double - at least as big as double
- Range of exponents for all 3 types is at least -37 to +37
- Cout
  - drops 0 after decimal point
  - cout.setf() overrides that behavior, it will display 6 decimal places
- If you want a constant to be a type float, use F or f suffix
- C++ guarantees 6 significant figures for float and 13 for double

- Arithmetic operators
  - /, *, +, -, %
  - Two values = operands
  - Module can't be used with floating point values -> compile time error
- Operator precedence and associativity
  - Arithmetic operators follow algebraic precedence
  - Use parentheses to enforce your own priorities
  - Associativity - if two operators share the same operand: 120 / 4 * 5
    - Multiplication and division - left to right
  - Division - if both operands are integers, fractional part of the result is discarded
- Type conversion
  - C++ converts value when
    - You assign a value of one arithmetic type to a variable of another arithmetic type
    - You combine mixed types in expressions
    - You pass argument to functions
  - If you convert bigger type to smaller type - behavior is undefined by C++

  - Truncation - discarding the fractional part (not rounding)
  - Integral promotions
    - When C++ evaluates expressions, it converts bool, char, unsigned char and short to int
      - Chicken and duck are converted to int -> sum -> convert result to int

- Conversions in passing arguments
  - It is possible to waive prototype control for argument passing
    - In that case C++ applies the integral promotion to the char, short and float

- Type cast
  - forms
    - (long) thorn - C
    - long (thorn) - C++
    - static_cast<long> (thorn)
  - First it cast value then it performs operations