UNIVERSITY OF MASSACHUSETTS
DARTMOUTH

ECE160: Foundations of Computer Engineering I

Lecture #5 – Constants

Instructor: Dr. Liudong Xing
SENG-213C, lxing@umassd.edu
ECE Dept .

Administrative Issues (1/30, Mon.)

- Homework #1 due **TODAY**
  - Please follow the "submission guidelines" available in the course website to submit your answers to your name folder at the class M: drive if you haven't
  - Late submission is subject to penalty.

- Lab#2 assigned today
  - Due by **5pm, Wednesday, Feb. 1**

- Lab#1 grade is available from M: drive
  - Please send your grade concern to TA Peter (glv@umassd.edu) and cc to me if there is any

Review of Lecture #4 (1)

- Four standard data types in C
  - void
  - short int, int, long int (integer)
  - char
  - float, double, long double (floating point)
- Variables
  - Declaration: to name a variable
  - Definition: to create a variable, and reserve memory for it
  - Initialization: assign an initial value to a variable

Review of Lecture #4 (2)

- Good programming practice
  - Have one definition/declaration per line
  - Have one initialization per line
  - Separate variable declaration from variable initialization
  - Always remember to initialize variables

```
int num1=0, num2=0;
double float1=0;
char chara1='a';
```

vs.

```
int num1;
int num2;
double float1;
char chara1;
num1 = 0;
num2 = 0;
float1 = 0;
chara1 = 'a';
```

Recommended!

Constants

- Different types of constants
- How to use constants in the program

Textbook: Chapter 2.3

Constants

- Constants are data whose values cannot be changed while the program is running.

- Constants have a type (like variables)
  - Integer constants
  - Floating point constants
  - Character constants
  - String constants

Integer Constants

- int temperature = 78; /* 78 is an integer constant*/

- Integer constants are simply coded as you would use them in everyday life!
- The default type is signed integer or signed long integer if the number is large
- You can override the default by specifying u or U (for unsigned) and l or L (for long) after the number

Integer Constants (Cont'd)

- Examples:

-32271L	long int
-100	int
78	int
76542LU	unsigned long int

- Note: there is no way to specify a short int constant in C!
- Good programming practice
  - Both upper- and lower- case codes are allowed, but
  - Prefer the capital notations L and U.

Floating-Point Constants

- These are numbers with decimal parts.
- The default form is double.
- If you want the resulting type to be float or long double use f or F, and l or L respectively.

- Again, always prefer capitals.

Floating-Point Constants (Cont'd)

- Examples:

0.0	double
2.3	double
3.1415926536L	long double
3.14	double
-2.58F	float

Character Constants

- Character constants are enclosed between two single quotes.
  - Examples:

'a'

'b'

- In addition, there can be a backslash \ (the escape character). The backslash is used when the character does not have a graphic associated with it.

backspace: '\b'

newline: '\n'

tab: '\t'

null character: '\0'

single quote: '\''

double quote: '\"'

String Constants

- A sequence of characters enclosed in double quotes.

"Hello"

"Hello\n"

- Difference between the null character and the empty string.

'\0' → null character

" " → empty string

1) A null character corresponds to 8 zero bits:
https://www.ascii-code.com/
2) An empty string is a string containing nothing.

Constants

- ✓ Different types of constants
- **How to use constants in the program**

How do we use constants in the program?

- Three different ways:
  - Literal constants
  - Defined constants
  - Memory constants

Literal Constants

- A literal is an unnamed constant used to specify data

- Examples:

```
b + 3.1   /*numeric literal 3.1*/
b * 2     /*numeric literal 2*/
'a'       /*a character literal */
"hello"   /*a string literal*/
```

Defined Constants

- Use preprocessor command:

  #define name expression

- Example:

  #define PI 3.14159

  #define num 2

- The expression that follows the name replaces the name wherever it is found in the source program

- Good programming practice:
  - Put all the #define statements at the beginning of your program.
  - Use the #define statement, instead of literal constants.

Memory Constants

- Use a C type qualifier:

  const type identifier = value;

- Example:

  const float PI = 3.14159;

- Memory constants fix the contents of a memory location.
- Can we change the value of PI in our program?

No!!!

An Example

- Write a C program that can output the value of PI

```c
#include "stdio.h"

void main(void)
{
   printf("The value of PI is: %f\n", 3.14159);
}
```

- Modify the program by changing the Literal Constant 3.14159 to a **Defined Constant**

- Modify the program by changing the Literal Constant 3.14159 to a **Memory Constant**

Summary of Lecture #5

- Constants
  - Four types: integer, floating point, character, string
  - Three ways to code constants in the program:
    - Literal: an unnamed constant used to specify data
    - Defined: use the preprocessor command #define name expression (the expression that follows the name in the command replaces the name wherever it is found in the source program)
    - Memory: use a C type qualifier: const type identifier = value; (memory constants fix the content of a memory location)

Things To Do

- Lab#2
  - Due by **5pm, Wednesday, Feb. 1.**

Next Topic

- Formatted Input/Output