Coding Guidelines

University of Massachusetts at Lowell

Instructor: Joe Russell
Website: http://www.geocities.com/joe_programming

Coding Guidelines

These guidelines are applicable to the following classes:

16.216 ECE Applications Programming
16.453 Software Engineering (Undergraduate)
16.553 Software Engineering (Graduate)
90.212 Introduction to Programming with C – Part II
90.267 C Programming
90.268 C++ Programming
90.364 Problem Solving with C

Commenting

- The following require proper commenting
  - Function Description Headers
    - Every function you write should have a function description header above the function definition
  - Ambiguous Variable Names
    - Names that are not descriptive
  - Parts of code that are not easily understood by another programmer
    - Perhaps a complicated equation
    - Perhaps a change in algorithm that needs explaining

Function Description Header using C++ Commenting

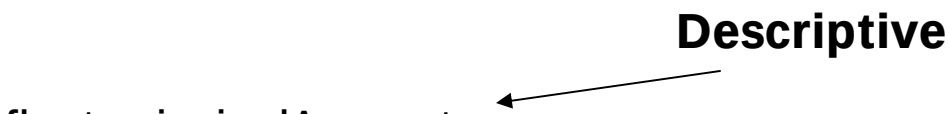
```
////////////////////////////////////////////////////////////////////////////
//          calculateInterest()
//
//          Description:  This function will return the interest
//          on a loan or investment for a given period of time.
//
//          Inputs:     principalAmount
//                      numberMonths
//
//          Outputs:    none
//
//          Return:     interest
////////////////////////////////////////////////////////////////////////////
```

Function Description Header using C Commenting

```
/*************************************************************
 *          calculateInterest()
 *
 *          Description:  This function will return the interest
 *          on a loan or investment for a given period of time.
 *
 *          Inputs:     principalAmount
 *                      numberMonths
 *
 *          Outputs:    none
 *
 *          Return:     interest
*************************************************************/
```

Commenting Ambiguous Variable Names

- Ambiguous Variables should always have a comment to describe their usage.
- No commenting is necessary if the variable names are descriptive




```
float principalAmount;

int i, j, k;   // Counter Variables

int i, j, k;   /* Counter Variables */
```

Commenting Parts of Code Not Easily Understood

- In the code snippet below, we assume that the hardware is designed to support our LED bit definitions.

```
// Set status LED color to green
*pLedRegister &= ~0x30;
*pLedRegister |= 0x20;
```

Note: There are better ways to do the above code using bit fields and/or bit masks. We will discuss these later in the semester.

White Space

- Make good use of white space for program clarity
  - Separate function definitions by two lines
  - Separate logical blocks of code by one line
- Do not have multiple program statements on the same line

Curly Braces

- The open curly brace { should exist on a line all by itself
- The closing curly brace } should exist on a line all by itself

Indentation

- Code that exists between and open and closed curly brace should be indented by one tab (or 4 single spaces)
- When optionally choosing not to use curly braces (e.g., single statement after an if statement), the code should be indented by one tab (or 4 single spaces)

Naming Functions

- All Function names must be Descriptive
- All user-defined function names must be written in Bumpy-Cased Notation
  - With Bumpy-Cased Notation, word boundaries are capitalized. Capitalizing the first letter is optional.
  - Below are some function prototype examples

```
float calculateInterest(float amt, float rate, float time);

int menu(void);

int GetUserInput(int *pChoice1, int *pChoice2);
```

Naming Variables

- All Variable names must be Descriptive
- All user-defined variables names must be written in Bumpy-Cased Notation
  - With Bumpy-Cased Notation, word boundaries are capitalized. The first letter should NOT be capitalized.
  - Below are some variable declaration examples

```
char userInput[80];

int xPosition, yPosition, zPosition;

char *pString = NULL;
```

Naming Symbolic Constants

- Symbolic Constants must be Capitalized and Descriptive
- Great care must be paid to usage of parenthesis

```
#define DAYS_PER_WEEK 7
#define INPUT_BUFFER_SIZE 80

#define BASE_ADDRESS 0x10000000
#define STATUS_REG  ((BASE_ADDRESS) + 0xA0B0)

#define ERR_MSG_1 “ERROR: Could not Allocate Memory\n”
#define ERR_MSG_2 “ERROR: Could not Open File\n”
```

Naming Macros

- Macros must be Capitalized and Descriptive
- Great care must be paid to usage of parenthesis

```
#define TOGGLE(x) (!(x))

#define MAX(x,y) ((x) > (y) ? (x) : (y))
```