Intro to C

C is for cookie, that's good enough for me...

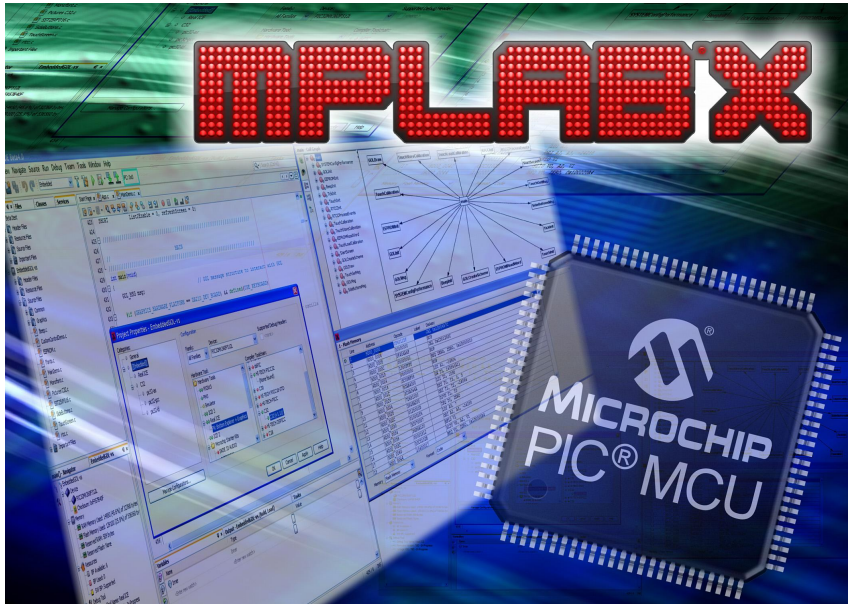
C Programming Language

- Created in 1972
- Many languages were built on C
  - C++, Java, Objective-C

- Many languages were built on languages that trace roots back to C
  - C#, Javascript, Matlab, Python

Fire up MPLABx

- Make a new project
- PIC18F4520
- Simulator
- Create a .c file from the template



Comments

```
// Single line comment

/*
   Comment out
    multiple
      lines
*/
```

```
/** Global Variables ***********************************************/
int sampleVariable = 0;

/*********************************************************************
 * Function:        void main(void)
 *********************************************************************/
#pragma code

void main(void) {
    // Set the clock to 4 MHz
    OSCCONbits.IRCF2 = 1;
    OSCCONbits.IRCF1 = 1;
    OSCCONbits.IRCF0 = 0;

    // Pin IO Setup
    OpenADC(ADC_FOSC_8 & ADC_RIGHT_JUST & ADC_12_TAD,
            ADC_CH0 & ADC_INT_OFF & ADC_REF_VDD_VSS,
            0x0B); // Four analog pins
    TRISA = 0xFF; // All of PORTA input
    TRISB = 0xFF; // All of PORTB input
    TRISC = 0x00; // All of PORTC output
    TRISD = 0x00; // All of PORTD output
    PORTC = 0x00; // Turn off all 8 Port C LEDs

    // This area happens once
    // Good for initializing and things that need to happen once

    while (1) {
        // This area loops forever
    }
}
```

Preprocessor directives (things with #)

Not really our focus today (but **very** necessary). If it starts in a #, then it doesn't end in a **;**

```
/** Header Files ************************************************/
#include <p18f4520.h>
#include <stdio.h>
#include <delays.h>
#include <adc.h>

/** Configuration Bits ******************************************/
#pragma config OSC = INTIO67
#pragma config WDT = OFF
#pragma config LVP = OFF
#pragma config BOREN = OFF
#pragma config XINST = OFF

/** Define Constants Here ***************************************/
#define SAMPLE 100
```

Code starts with the **main** function

```
void main(void) {
    // Set the clock to 4 MHz
    OSCCONbits.IRCF2 = 1;
    OSCCONbits.IRCF1 = 1;
    OSCCONbits.IRCF0 = 0;

    // Pin IO Setup
    OpenADC(ADC_FOSC_8 & ADC_RIGHT_JUST & ADC_12_TAD,
            ADC_CH0 & ADC_INT_OFF & ADC_REF_VDD_VSS,
            0x0B); // Four analog pins
    TRISA = 0xFF; // All of PORTA input
    TRISB = 0xFF; // All of PORTB input
    TRISC = 0x00; // All of PORTC output
    TRISD = 0x00; // All of PORTD output
    PORTC = 0x00; // Turn off all 8 Port C LEDs

    // This area happens once
    // Good for initializing and things that need to happen once

    while (1) {
        // This area loops forever
    }
}
```

Declaring variables is required in C

- Declaring variables

```
// Just declaring variables
char x;
unsigned char y;
int Dave, Steve, a, b, c, d, e, f, g, h;
long Bob;
float Steve;
char arrayBob[10];
```

- Initializing (defining) a variable at declaration

```
// Initialized data
int x = 0;
float f = 1.253;
char message1[] = "Handy string syntax"
int numlist[] = {1, 1, 2, 3, 5, 8, 13, 21, 34};
```

Global vs local variables

(variable scope)

Outside any function. Visible to the entire .c file (module).

```
/** Global Variables ************************************************/
int sampleVariable = 0;
```

Within a function (first thing). Visible within function.

```
void main(void) {
    int localVariableInMain = 7;

    // This area happens once
    // Good for initializing and things that need to happen once

    while (1) {
        // This area loops forever
    }
}
```

Variable scope details for later reference

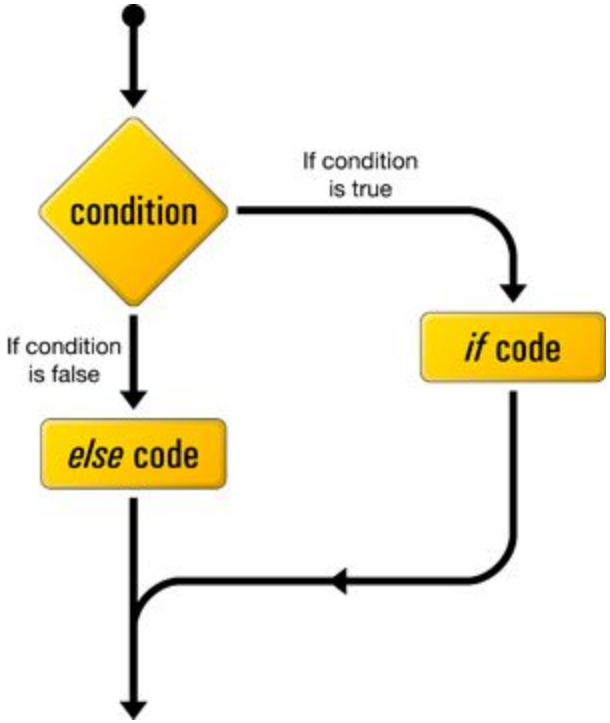
Global variables (Module level variables)

- Defined outside of any function (within Global variables section)
- Variable can be used anywhere within the .c file
- Variable does not need to be passed to functions
- Excellent for interrupts (explain interrupts later in course)

Local variables

- Defined within functions (must always FIRST thing in function!)
  - Causes a syntax error if you try declaring variable mid function
- Need to be passed by functions
- Helps organize your code
- Better programming practice BUT the debugger sometimes doesn't handle well

Control structures - if and switch



Control structures

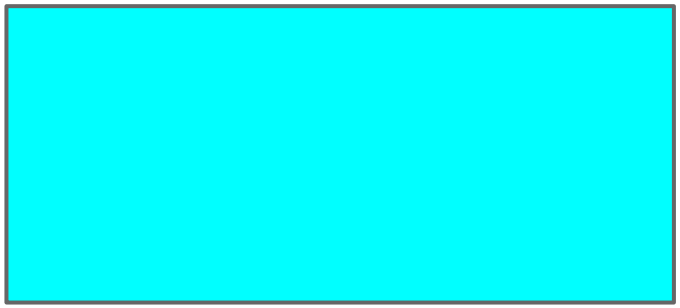
- Concepts you know, syntax is new
- if conditional statement
  - switch conditional statement (may be new)
- Next time
  - for loops
  - while loops

if – Standard if else structure

```
// if - else statement
if (Bob == 12)
{
     Dave = 5;
     PORTB = 0x03;
}
else
{
     Dave = 3;
     PORTB = 0x01;
}
```

if – Beware common errors

```
// Basic if statement
if (Bob == 10)
    Dave = 5;
```



if – Beware common errors

```
// BAD if statement
if(Bob == 10);
    Dave = 5;
```

```
// Multiple lines if statment
if (Bob == 11)
{
    Dave = 5;
    PORTB = 0x03;
}

// BAD multipe lines if statement
if (Bob == 11)
    Dave = 5;
    PORTB = 0x03;
```

if – Your turn

- Make a variable **age**
  - char age = 10;
- Make an if-else statement that prints
  - “age = %d which is greater than 15\n” or
  - “age = %d which is less than or equal to 15\n”
- Then change the variable age to **20**
  - Rerun the program

if – Multiple Conditions

```
// Multiple conditions
if (   (Bob == 11) && (x < 3)   )
{
     Dave = 5;
     PORTB = 0x03;
}

// BAD statement
if ( 2 < x < 6 )
{
     Dave = 5;
     PORTB = 0x03;
}


// Conditional Statements == != < <= > >=
// Logical Operators      && ||
```

else if – Runs first match only

```
// if - else if - else if - else statement
if (Bob == 12)
{
     Dave = 5;
     PORTB = 0x03;
}
else if (Bob == 13)
{
     Dave = 4;
     PORTB = 0x02;
}
else
{
     Dave = 3;
     PORTB = 0x01;
}
```

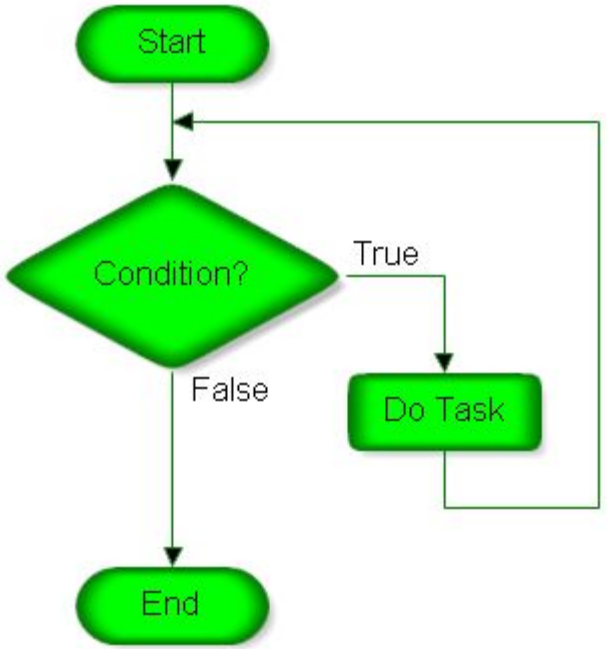
switch – Same as else if (using only ==)

```
// Switch statement
switch (Bob)
{
     case 3:
          Dave = 3;
          break;
     case 4:
          Dave = 4;
          break;
     case 5:
          Dave = 50;
          break;
     default:
          Dave = 0;
          break;
}
```

Convenience function with weird syntax. Handy for discrete value conditions

Can ALWAYS be done with an else if series of statements. :(

Loops - for and while



for

```
int i;

for ( i=0 ; i<10 ; i=i+1)
{
    printf("i = %d\n", i);
}
```

```
for ( initialization ; conditional statement ; update variable statement )
{


}
```

for – Standard shortcut

```
int i;

for ( i=0 ; i<10 ; i=i+1)
{
    printf("i = %d\n", i);
}

for ( i=0 ; i<10 ; i++)
{
    printf("i = %d\n", i);
}
```

Basic for loop

Same loop using shortcut operator

Shortcut only useful for *for* loops that increment the variable by 1. Very common though.

for – Your turn

- Write a for loop
  - runs from **age** = 20 to **age** = 0
  - decrements **age** by 5 each time
  - i.e. **age** should be 20, 15, 10, 5, and 0
- Inside the for loop, put an "if"-"else if"–"else" that prints (as appropriate)
  - age = %d which is greater than 15
  - age = %d which is equal to 15
  - age = %d which is less than 15

while

```
// A while loop that runs forever
while (1)
{
    // This area loops forever
}
```

```
// A while loop implementing a for loop
Bob = 20;
while ( Bob < 35)
{
    printf("Bob is %d years old",Bob);
    Bob++;
}
```

Functions (not today)

```
/** Local Function Prototypes ***************************************/
void sampleFunction(void);



/*******************************************************************
* Additional Helper Functions
*******************************************************************/

/*******************************************************************
* Function:          void sample(void)
* Input Variables:   none
* Output Return:     none
* Overview:          Use a comment block like this before functions
*******************************************************************/
void sampleFunction()
{
    // Some function that does a specific task
}
```

C programming references

- CSSE120 videos (uses Eclipse not MPLABx)
  - http://www.youtube.com/watch?v=iT1Z7p7OFYI&list=PLBK7yyieyrAYzgXY9nXwl_5CYQvnzpWCp
- Essential C: An introduction
  - http://cslibrary.stanford.edu/101/EssentialC.pdf
- Programming in C (4th Edition) – Steve Kochan
- Thousands more on the web
  - Plenty of time since 1972 for posts!

References

http://muppet.wikia.com/wiki/C_is_for_Cookie

http://ucan.us/doyetech/images/if-then-else-flowchart.png

http://www.rff.com/flowchart_structure_loop.png