ChipKit/Arduino 0023 C/C++ Reference Sheet 0.1

Structure
//Declarations
void setup () { ... }
void loop () { ... }

Pointer Access
& reference operator
* dereference operator

Bitwise Operators
& (bitwise and) | (bitwise or)
^(bitwise xor) ~(bitwise not)
<< (bitshift left) >> (bitshift right)

Control Structures
```
if (xx > 10) { ... } else { ... }
switch (myvar)
{
  case 1:
   break;
  case 2:
   break;
 default;
}
for (int ii = 0; ii <=255; ii++) { ...}
while (xx < 10) { ... }
do { ... } while ( x < 10);
continue; //Goto next in do/for/while loop
return xx; //Or 'return:' for voids
```

Further Syntax
// (Single line comment)
/* (multi-line comment) */
#define LEDS 7 //No semi colon
#include <IOShieldOled.h>

General Operators
= (assignment operator)
+ (addition) - (subtraction)
* (multiplicaiton) / (division)
% (modulo)
== (equal to) != (not equal to)
< (less than) > (greater than)
<= (less than or equal to)
>= (greater than or equal to)
&& (and) || (or) !(not)

Constants
HIGH|LOW
INPUT|OUTPUT
true|false
143 //Decimal number
0173 //Octal number
0b11011111 //Binary
0x7B //Hex number
7U //Force unsigned
15UL //Force Unsigned long
10.0 //Forces floating point
2.4e5 // 240000

Data Types
void
boolean (0,1, false, true)
char (e.g. 'a'-128 to 127)
unsigned char (0 to 255)
byte (0 to 255)
int (-32,768 to 32,767)
unsigned int (0 to 65,535)
word (0 to 65535)
long (-2,146,483,648 to 2,147,483,647)
unsigned long (0 to 4,294,967,285)
float (-3.4028236E+38 to 3.4028235E+38)
double (currently same as float)
sizeof(myint) //returns 2 bytes

Qualifiers
static //persists between calls
volatile //use RAM (nice for ISR)
const //make read only
PROGMEM //use flash

Compount Operators
++ (increment) -- (decrement)
+= (compound addition)
-= (compond subtraction)
*= (compund multiplicaiton)
/= (compound division)
&= (compound bitwise and)
|=(compund bitwise or)

Strings
```
char s1[15];
char s2[8]={'c','h','i','p','k','i','t'};
char s3[8]= {'c','h','i','p','k','i','t','\0'};
//Above includes null termination
char s4[] = "chipkit";
char s5[8] = "chipkit";
char s6[15] = "chipkit";
```

Arrays
```
int myInts[6];
int myPins[] = {2,4,8,6,7};
int mySensVals[6] = {2,4,-8,3,2};
```

Conversion
char() byte()
int() word()
long() float()

Time
unsigned long millis() //50 days overflow
unsigned long micros() //70 min overflow
delay(ms)
delayMicroseconds(us)

Math
min(xx, yy) max(xx, yy) abs(xx)
constrain(xx, minval, maxval)
map(val, fromL, fromH, toL, toH)
pow(base, exponent) sqrt(x)
sin(rad) cos(rad) tan(rad)

Random Numbers
randomSeed(seed) //long or int
long random(max)
long random(min,max)

Bits and Bytes
lowByte() highByte()
bitRead(x, bitn) bitWrite(x, butn, but)
bitSet(x, bitn) bitClear(x, bitn)
bit(bitn) //bitn: -0LSB 7-MSB

External Interrupts
attachInterrupt(interrupt, function, [LOW | CHANGE| RISING| FALLING])
detachInterrupt(interrupt)
interrupts()
noInterupts()

Adanced I/O
tone(pin, freqhz)
ton(pin,freqhz,duration_ms)
noTOne(pin)
shiftOut(dataPin, clockPin, [MSBFIRST, LSBFIRST], value)
unsigned long pulseIn(pin, [HIGH | LOW])

Digital I/O
pinMode(pin, [INPUT | OUTPUT]));
digitalWrite(pin, value)
int digitalRead(pin)
//Write HIGH to inputs to use pull-up resistors

Analog I/O
analogReference([DEFAULT | INTERNAL | EXTERNAL]);
analogWrite(pin, value) //PWM
int analogRead(pin) //Call twice if swithcing pins from high z Source

Serial (communicate with PC or via RX/TX)
begin([300,1200,2400,4800,9600,1440,19200,28800,3840057600,115200]);
end()
int available()
int read()
flush()
print()
println()
write()

IOShield EEPROM
readString(unit16_t address, unit8_t *sz, int size)
readString(unit16_t address, char *sz, int size)
write(uint16_t address, uint8_t data)
writeString(uint16_t address, uint8_t *sz, int size)
writeString(uint16_t address, char *sz, int size)
\writeString(uint16_t address, char *sz)

IOSHield Temperature
config(uint8_t configuration)
float getTemp()
setTempHyst(float tMin)
setTempLimit(float_t tMax)
float convCtoF(float tempC)
float convFtoC(float tempF)

IO Shield Libraries
IOShield OLED
begin()
end()
displayOn()
displayOff()
clear()
clearBuffer()
updateDisplay()
setCursor(int xch, int ych)
getCursor(int *pxch, int *pych)
defineUserChar(char ch, BYTE *pbDef
setCharUpdate(int f)
int getCharUpdate()
putChar(char ch)
putString(char *sz)
setDrawColor(Byte chlr)
setDrawMode(int mod)
getDrawMode()
BYTE* getStdPattern(int ipat)
setFillPattern(BYTE *pbPat)
moveTo(int xco, int yco)
getPos(int *pxco, int *pyco)
drawPixel()
BYTE getPixel()
drawLine(int xco, int yco)
drawRect(int xco, int yco)
getBmp(int dxco, int dyco, BYTE *pbBmp)
putBmp(int dxco, int dyco, BYTE *pbBmp)
drawChar(char ch)
drawString(char *sz)