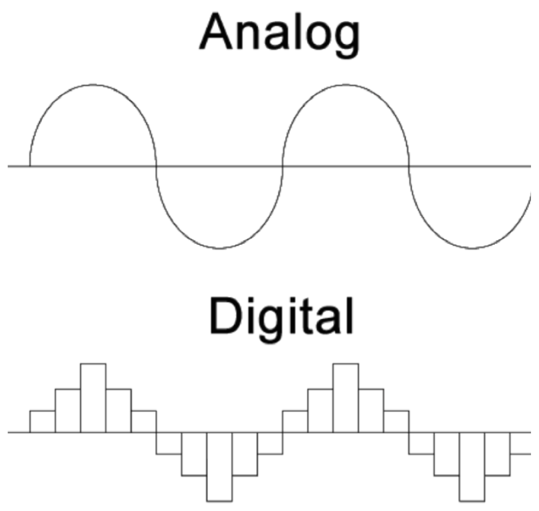
RedBoard Reference Sheet

Analog:
Information is translated into electric pulses of varying amplitude.
range of values; organic
(example: joystick is analog input)

Digital:
Translation of information is into binary format (zero or one) where each bit is representative of two distinct amplitudes.
on or off
(example: buttons are digital input)



General Data Types:

Keyword	Data Type	Description	Example	size (bytes)
bool	boolean	true or false	bool isRight = true;	4
int	integer	whole number (no decimal or fraction)	int numCats = 25;	4
float	floating point	number precise to 6 decimal places	float weight = 15.6;	4
double	floating point	number precise to 15 digits (decimal)	double cash = 23.4;	8
char	character	a single number, letter, or symbol	char charA = 65;	1

*to check the size of a data type use: sizeof(datatype);
*char is usually used for a character pointer or for an array of characters

Data Type Modifiers:

Keyword	Description
signed	can be positive or negative
unsigned	can only be positive
short	shorter range
long	longer range

*modifiers can be combined.
(example: unsigned short int)

Examples:

Example	Typical Range	size
signed int	-2147483648 to 2147483647	4
unsigned int	0 to 4294967295	4
short int	-32768 to 32767	2
long int	-2147483648 to 2147483647	4

Operator Precedence Chart:
http://en.cppreference.com/w/cpp/language/operator_precedence

Function Syntax:
returnType functionName (parameter1, parameter2, ...) { statements }

Example:
int TotalCatFood(int numCats, float catFoodUnit) {return numCats * (int) catFoodUnit;}

RedBoard Reference Sheet

<u>Conditional Statements</u>:

if statement	**if, else if, else statements**	**ternary expression**	**switch statement**

if (condition)
{
  statements;
}
	if (condition1)
	{
	  statements1;
	}
	else if (condition2)
	{
	  statements2;
	}
	else
	{
	  statements3;
	}
		a ? b : c
		
		reads as:
		if (a)
		  b;
		else
		  c;
			switch (variable)
			{
			 case value:
			  statements1;
			  break;
			 case value2:
			  statements2;
			  break;
			 default:
			  statements3;
			  break;
			}

Examples:			

int i = 0;
 	
if (i < 5)
{
 printf("hi");
}
	int i = 5;
	
	if (i < 2)
	{
	 printf("i is less than 2");
	}
	else if (i < 5)
	{
	 printf("i is less than 5");
	}
	else
	{
	 printf("i is not less than 5");
	}
		int a = 0;
		int b = 5;
		
		a = (a == b) ? a : b;
		
		//if a is equal to b
		 //set a's value to a's
		//otherwise
		 //set a's value to b's
		
		printf("a is %d", a);
		
		//print value of a
			int buttons = 3;
			switch (buttons)
			{
			 case 1:
			  printf("1");
			  break;
			 case 2:
			  printf("2");
			  break;
			 case 3:
			  printf("3");
			  break;
			 default:
			  break;
			}

Output:			
hi	i is not less than 5	a is 5	3

*printf() prints to the command window – Serial.print() works somewhat similar to printf()

RedBoard Reference Sheet

Loops:

while loop	**do…while loop**	**for loop**

while (condition)
{
   statement(s);
}
	do
	{
	   statement(s);
	} while (condition);
		for (init; condition; increment)
		{
		   statement(s);
		}

Examples:		

int i = 5;
while (i > 0)
{
   printf("%d", i--);
}
// i is now 0
	int i = 0;
	do
	{
	   printf("%d", i--);
	} while (i > 0)
	// i is now -1
		for (int i = 5; i > 0; --i)
		{
		  printf("%d", i); //prints i
		}
		
		// i is now 0

Output:		
54321	0	54321

Loop Commands:		
Command:	**Explanation**:	**Example**:

continue;	Goes back to the top of the loop. Use this if you want to keep moving through the loop, but don't need to run the rest of the code that iteration.	//loop through all objects
		for (int j = 0; j < numObj; ++j)
		{
		//skip the rest of the loop & go back to top if not active
		  if (objList[j].active == false)
		  {
		    continue;
		   }
		
		  ++objList[j].value; //if active, increment value
		}

break;	Breaks out of the loop.	for (; ;) //infinite loop
		{
		  //leave loop when number of pelicans is equal to ducks'
		  if (pelican.count == duck.count)
		    break; //will break out of the loop
		
		    duckSystem();
		    pelicanSystem();
		   }

RedBoard Reference Sheet

<u>Functions/Methods to Know</u> (specific to Arduino sketch):

Common Sketch Functions	
Function:	**Example**:
void delay(int milliseconds);	delay(1000); //pauses for 1000 milliseconds
void pinMode(int pin_number, int mode);	pinMode(13, OUTPUT); //sets pin 13 as an OUTPUT pin
void digitalWrite(int pin_number, int state); //state can be HIGH or LOW	digitalWrite(13, HIGH); //sets pin 13 to a HIGH state

int digitalRead(int pin_number);
//reads in a state that is either HIGH or LOW
	digitalRead(13);
	//reads the state of pin 13
	//(could be a button or switch)

void analogWrite(int pin_number, int value); //value should be between 0 and 255 (duty cycle for PWM)	analogWrite(11, 100); //sets pin 11 to speed 100
int analogRead(int pin_number); //pin should be analog and attached to an analog input device	analogRead(0); //reads input from analog pin 0

Common Serial Window Methods:		
Method:	**Explanation**:	**Example**:
Serial.begin(int baud_rate); //usually written in setup() function	begins communication between computer and board	Serial.begin(9600);

Serial.print(“text to print”);	prints text to the serial window; text in quotations appears exactly as typed	Serial.print(“hi”);
		//prints hi
		int value = 5;
		Serial.print(value);
		//prints 5

Serial.println("automatically moves to a new line after this statement");	similar to print(), except that it moves to a newline after itself	Serial.println(“hi”); /*prints hi then moves to a newline*/

Serial.available();	checks if the user has typed anything into the Serial window	if (Serial.available())
		  Serial.println(“hi”);

Serial.parseInt();	reads in an integer from the Serial window	int a;
		a = Serial.parseInt();

*For a full list of Serial Window Methods go to: https://www.arduino.cc/en/Reference/Serial

Common Servo Methods:	
Method:	**Example**:
Servo_object.attach(int pin_number);	servo1.attach(5);
Servo_object.detach();	servo1.detach();
Servo_object.write(int degrees); //turns servo to an exact degree	servo1.write(90); //turns servo1 to the 90 degree angle

*For a full list of Servo Methods go to: https://www.arduino.cc/en/Reference/Servo

Common Liquid Crystal Display Methods:	
Method:	**Example**:
lcd.begin(int charNum, int lineNum);	lcd.begin(16, 2); //2 lines of 16 characters
lcd.clear();	lcd.clear(); //clears the screen
lcd.print("text to print");	lcd.print("hello, world!"); //prints hello, world!
lcd.setCursor(int charNum, int lineNum);	lcd.setCursor(0,1); //moves cursor

*For a full list of LCD Methods go to: https://www.arduino.cc/en/Reference/LiquidCrystal

Miscellaneous Functions	
Function:	**Example**:
int random(int value); //picks a random number between 0 & (value-1)	random(7) //picks a random number between 0 and 6

int map(int old_value,
int old_range_min,
int old_range_max,
int new_range_min,
int new_range_max);
	int lightLevel = -5;
	lightLevel = map(lightLevel, 0, 1023, 0, 255);
	/*converts lightLevel into a new value based on it's placement in the old range*/

int constrain(int value,
int range_min,
int range_max);
	int lightLevel = -1;
	lightLevel = constrain(lightLevel, 0, 255);
	/*ensures lightLevel doesn't go under 0 or over 255*/

tone(buzzerPin, frequency, duration); //makes buzzer play a particular tone	tone(5, 262, 4); //plays C note on pin 5 for 4 ms
noTone(buzzerPin); //stops the buzzer from playing tones	noTone(5); //stops sound on pin 5

unsigned long millis();
//returns amount of time the program has run
//resets after 50 days
	unsigned long timeSinceStart = millis();
	//time is recorded in milliseconds

bitWrite(byte data, int desiredPin, int desiredState);	bitWrite(0, 0, HIGH) //writes 1 to bit 0 (rightmost side) on variable 0

shiftOut(datapin, clockpin, bitOrder, data);
//bitOrder is either MSBFIRST or LSBFIRST
//MSBFIRST: most significant bit first (left)
//LSBFIRST: least significant bit first (right)
	shiftOut(2, 3, MSBFIRST, 0);
	//on pin 2 increment most significant (leftmost)
	//bit first
	//toggle pin 3 once pin 2 has been set to 0