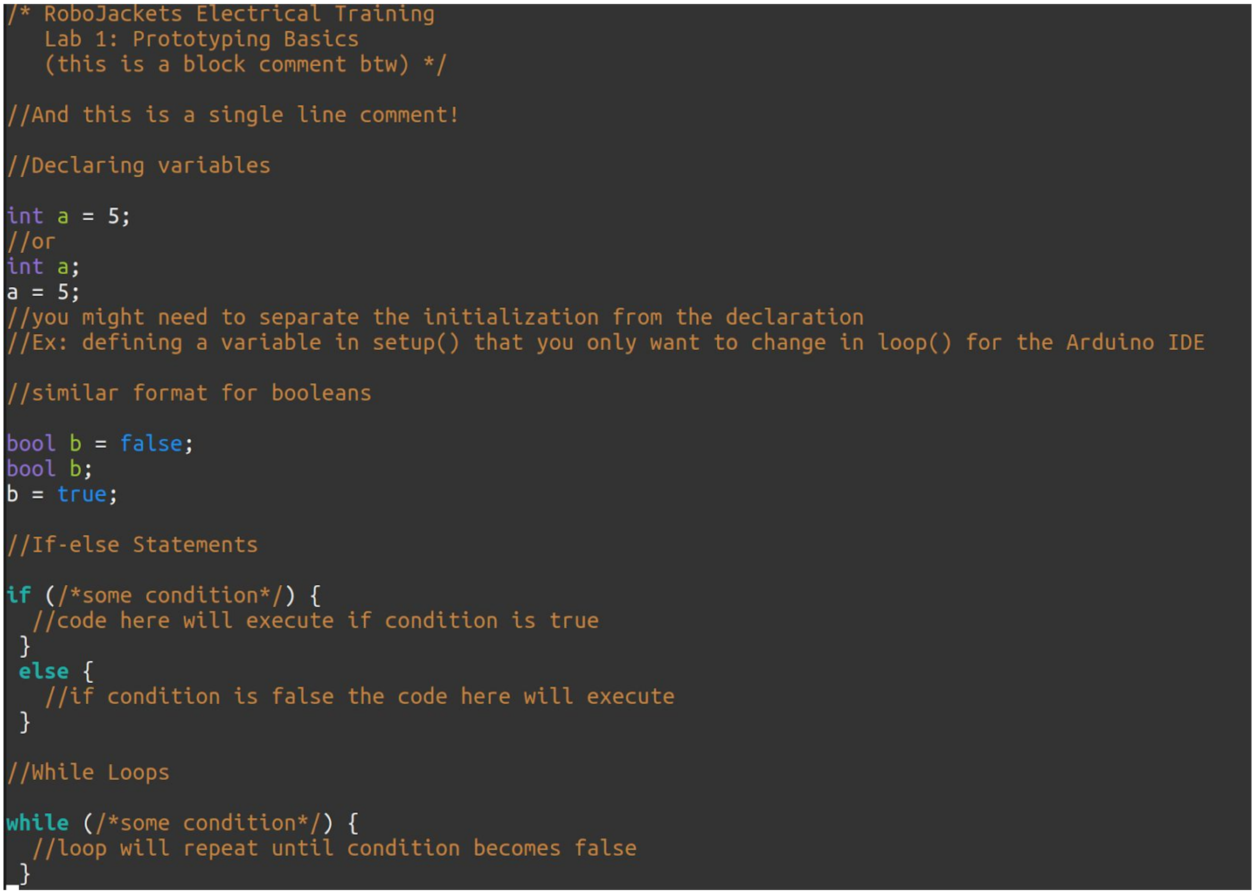
Lab 1

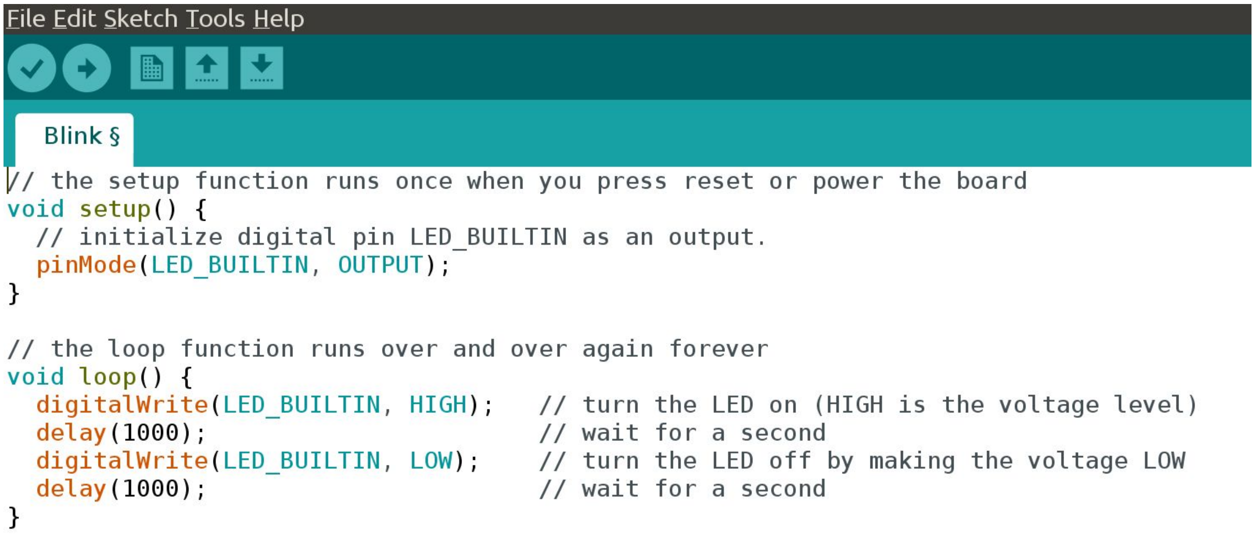
C++ Basics:

- To program Arduinos you will be writing C++ code within the IDE
- Some basics are outlined below



```
/* RoboJackets Electrical Training
   Lab 1: Prototyping Basics
   (this is a block comment btw) */

//And this is a single line comment!

//Declaring variables

int a = 5;
//or
int a;
a = 5;
//you might need to separate the initialization from the declaration
//Ex: defining a variable in setup() that you only want to change in loop() for the Arduino IDE

//similar format for booleans

bool b = false;
bool b;
b = true;

//If-else Statements

if (/*some condition*/) {
  //code here will execute if condition is true
 }
 else {
   //if condition is false the code here will execute
 }

//While Loops

while (/*some condition*/) {
  //loop will repeat until condition becomes false
 }
```

Arduino IDE:



```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                       // wait for a second
}
```

- Above is a screenshot from the Arduino IDE which is the application in which you will write your code and upload it to your Arduino
- This program will blink an LED on the Arduino itself
- A typical program is split into two parts: the setup() function and the loop() function:
    - Setup()
        - This function is called when the program starts. It'll run when you power on or reset the Arduino board
        - Here is where you will initialize variables, set pin modes, etc.
    - Loop()
        - This function loops consecutively so the code you place here will constantly run
- Some common functions you should know are outlined below:
    - pinMode(pin, mode): Configures the specified pin to behave either as an input or an output
        - Pin: the pin number
        - Mode: INPUT, OUTPUT
    - digitalWrite(pin, value): Write a high or a low value to a digital pin
        - Pin: the pin number
        - Value: HIGH, LOW
    - digitalRead(pin): Read a value (high or low) from a specified pin)
        - Pin: the pin number
    - delay(time): Pauses the program for a specified amount of time
        - Time: the time for the delay in ms
- The IDE:
    - The checkmark is the "Verify" button which will compile your code
    - The arrow is the "Upload" button which will compile and load your code on your Arduino

    - Before you compile make sure to select the port on your computer where your Arduino is connected by going to Tools>Port

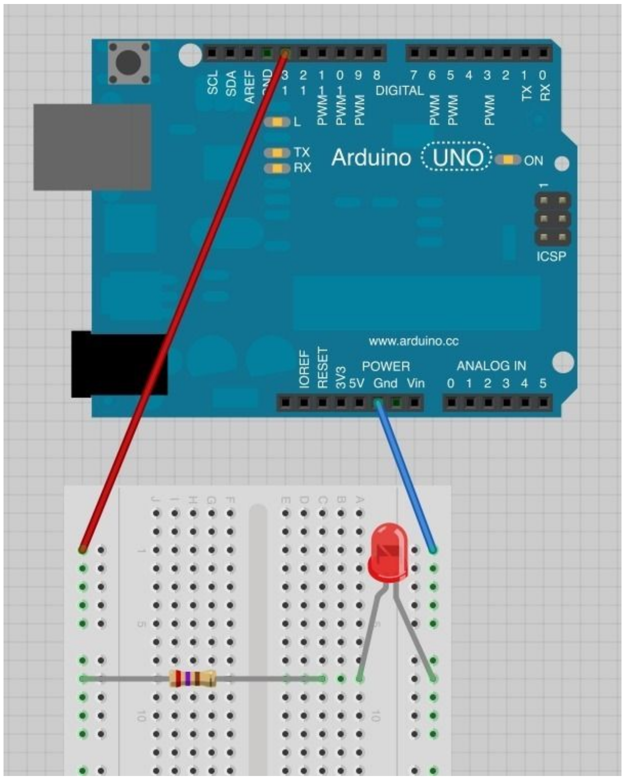
Uploading and Compiling an existing program:

- Here, we will be using the blinking LED example program that Arduino provides. This will blink an LED that exists on the Arduino itself. This is the same code that is pictured above

1) Accessing program: Files>Examples>Basics>Blink
2) Upload the program: Use the -> arrow to load the program onto the Arduino

Blinking an external LED on a breadboard:

1) Setup your breadboard in a similar style:
    - Remember to check the polarity of your LED (longer side is +)
2) Modify your code to blink the LED on the breadboard



Lab 1 Extended:

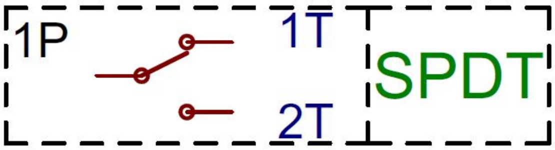
Pick some of the tasks to do (ranked in order of difficulty)

1) Use the provided switch as a digital input to the Arduino to control the LED
2) Create code to control 2 LEDs independently using the digital input of one switch
3) Use the digital input of the switch to select between two blink frequencies of an LED
4) Create code to implement a binary counter with three LEDs, using the digital input of a switch to iterate through each number (turning the switch on, then off, represents one count)

Background Material:

SPDT (Single Pole Double Throw) Switches

- These are switches with three terminals - one common terminal and two output terminals
- This common terminal (in the case of your switches, the middle black wire) can connect and switch to the two output terminals (the red and blue wires)


1P
1T
2T
SPDT