

# Board and IDE Connections

Now you know how to setup your Arduino board and upload code from the IDE. Right now, you just uploaded the default code. Next, let's look at the code and break down some of the key parts in order to develop an understanding of what each line does, and what can be changed.

Let's examine the *Blink* sketch that you just uploaded:

```
// the setup function runs once when you press reset or power
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH); // turn the LED on (HIGH is the positive voltage)
  delay(1000);            // wait for a second
  digitalWrite(13, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);            // wait for a second
}
```

The code comments (you'll learn more about this important concept later in the camp) in the **void setup()** function have some important information about connecting the software and hardware of the Arduino:

- `pinMode(13, OUTPUT);`
  - **pinMode** tells the Arduino some important information. The first number, 13, tells the Arduino that the code is going to interact with *something* that's on pin 13.
  - Recall that you attached an LED to pin 13. This line of code tells

the Arduino that there is something on this pin.

- The "OUTPUT" sets the pin to output. In this example, the output is the blinking LED.
- Important Troubleshooting: If you're having trouble with your LED not blinking, make sure that the pinMode number corresponds to the pin holding the LED!

Next, let's examine some of the key code in the **void loop()** function:

- **digitalWrite(13,HIGH);** and **digitalWrite(13,LOW);** are the commands that actually turn the LED on and off, resulting in the blinking.
  - Note that the number included in this must match the one in the pinMode() in the setup.
  - HIGH sends an on message to the LED whereas LOW sends an off message.
- **delay(1000);** results in a pause of 1 second. The values that delay takes are in milliseconds(ms) so 1000 is equal to 1 second.

## Important Concepts

---

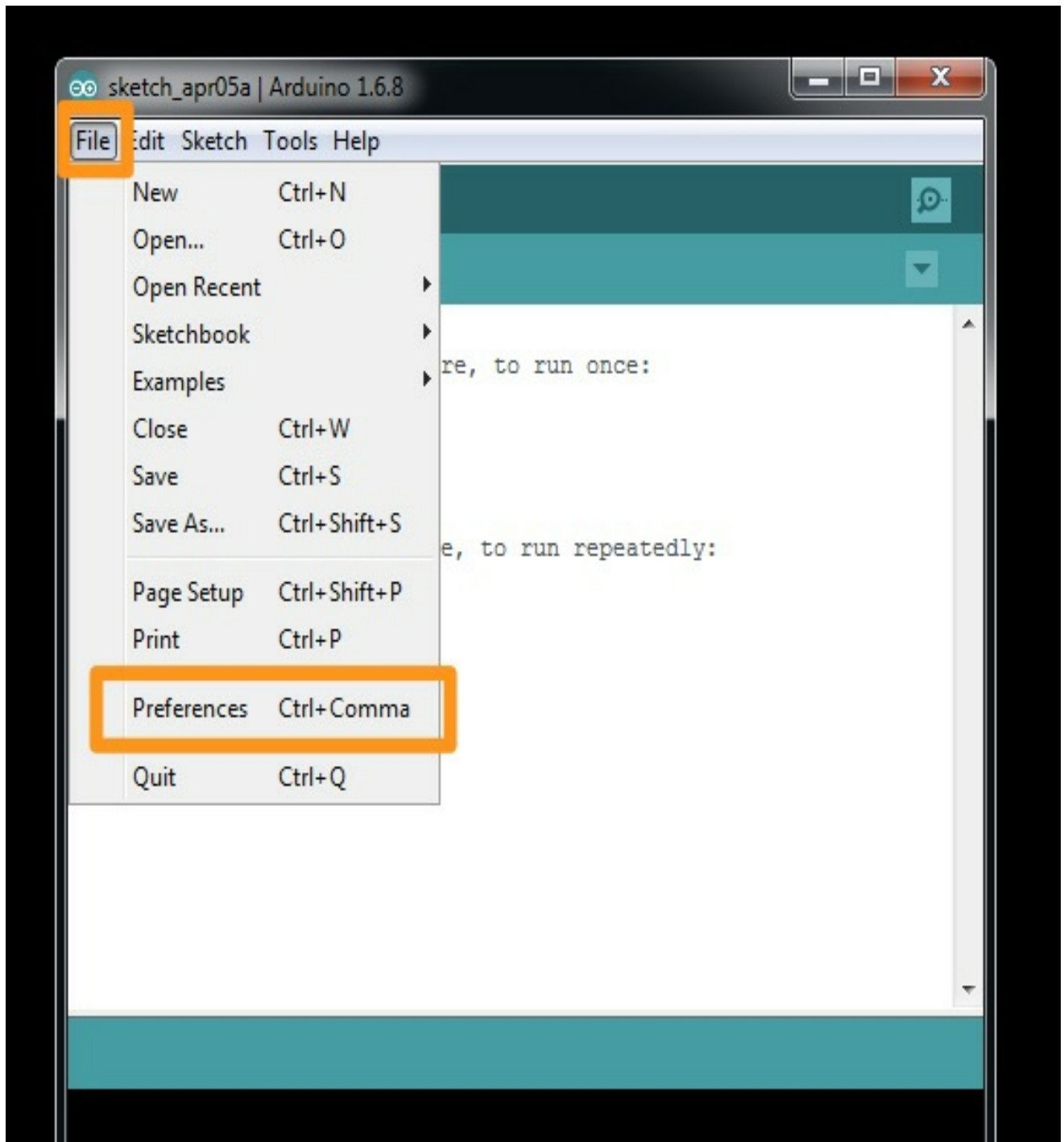
The comments in the code, as well as the explanations above, shed some light on the importance of making sure that your code correlates and matches the physical connections with your board.

When troubleshooting Arduino projects, and you'll likely have to do some troubleshooting in this camp, you have to check both the software and the hardware. If you build solid habits from the beginning, such as making sure that anything that references specific pins in the board have the correct number, you'll have a much easier time isolating and fixing issues!

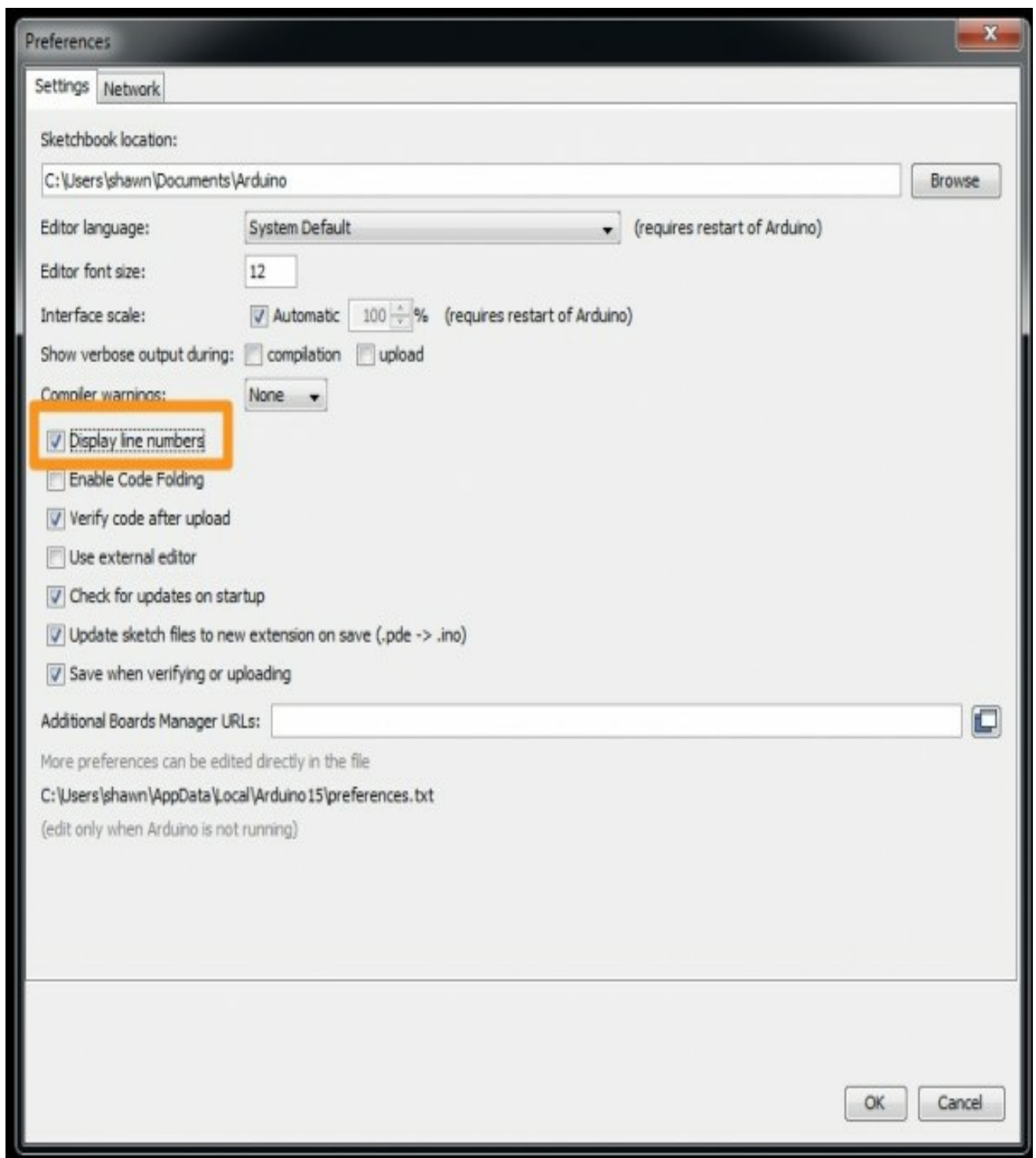
# Arduino Pro-Tip: Line Numbers

Here is a quick tip to help you when you are using the Arduino IDE. You want to be able to see the line numbers in the Arduino IDE to help track down bugs in error messages. To do so, just follow these steps:

1. Launch the Arduino IDE software
2. Click on the "File" menu and select "Preferences"

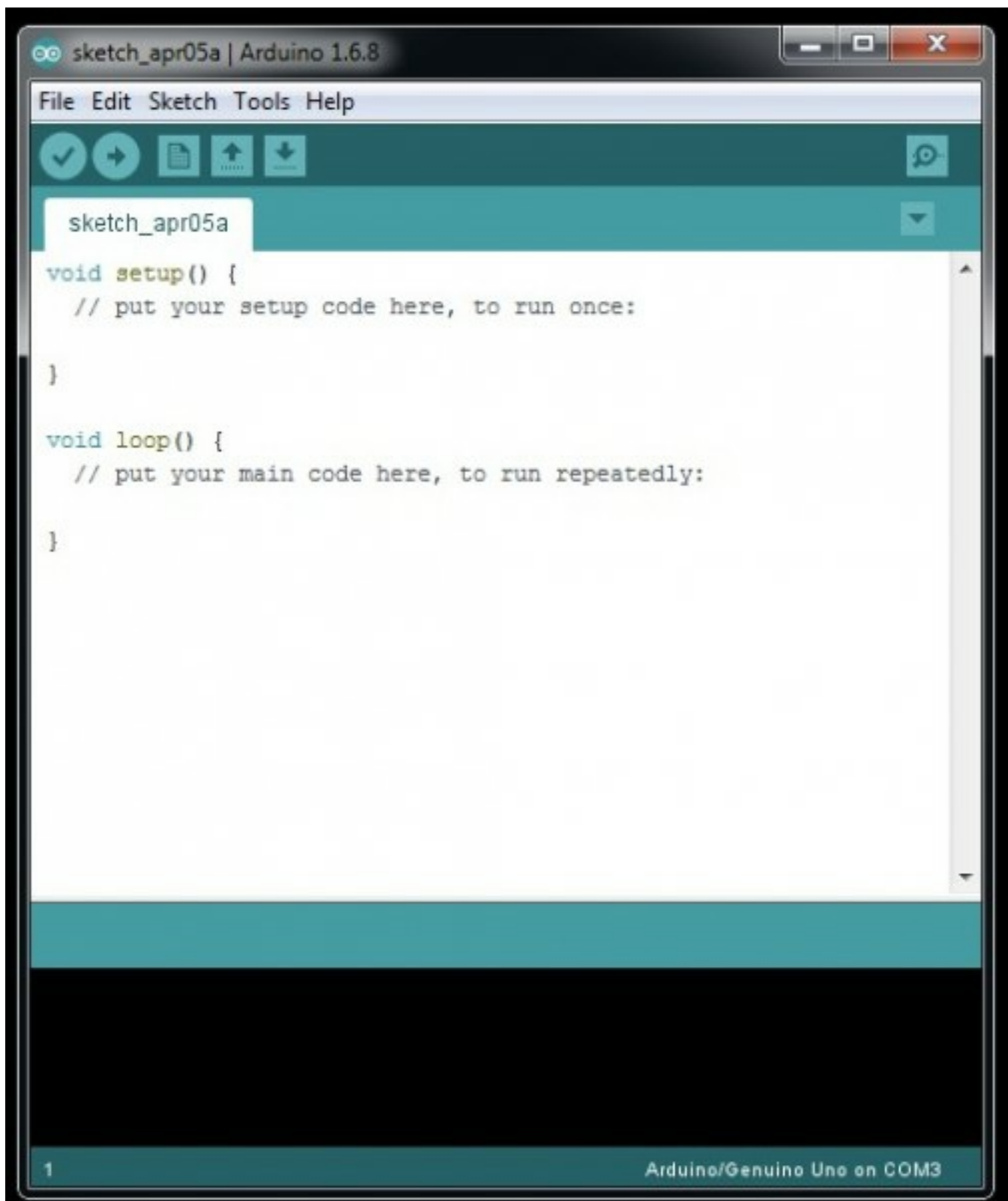


3. Now click on the check box next to "Display Line Numbers"

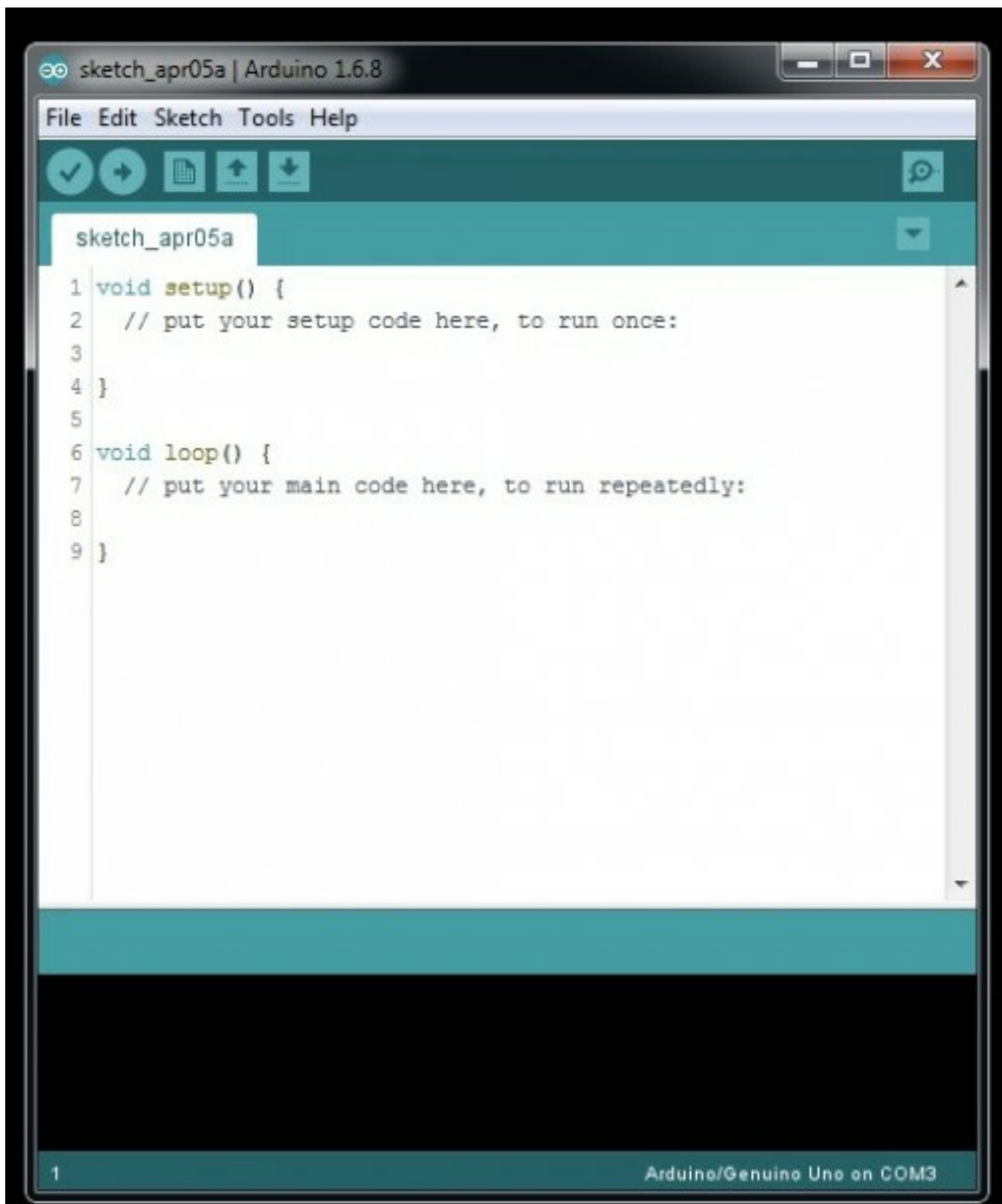


4. Click the "OK" button to close the preferences window.

Your code window will go from looking like this:



To looking like this with the line numbers down the left hand side:



Turning on the line numbers makes debugging and fixing your code much easier. Additionally, you'll be able to check to see if you're missing any code that you import since your line numbers won't match.