

Troubleshooting

C++ Tutorial

If you are using strings and get a weird compile error, make sure you have the following include command at the top of your file:

```
#include <string>
```

If you are using cout and get a weird compile error, make sure you have the following commands at the top of your file:

```
#include <iostream>
using namespace std;
```

If you get an **undefined** function **error**, make sure you have either:

- a) written its code before you call it or
- b) included a prototype at the top of the file.

A **function prototype** is just the return type, function name, and parameters followed by a semicolon.

Project Illuminate

To better **debug** your code, you can print information to the **Serial Monitor**. To do that, you have the functions `Serial.print()` and `Serial.println()`: they are the same, except that the second moves to a new line after it prints its argument. Each function will print either a string or a variable. For instance, to print `x = 5`, where 5 is the value of `x`, use the commands:

```
Serial.print("x =");
Serial.println(x);
```

To see the output, open the Serial Monitor from the Arduino IDE using the **Tools->Serial Monitor** menu options. When the Arduino IDE encounters an error, it will print the details to the **terminal** from which you launched the application. Unless the IDE also pops up a box or displays an error in the console (the black box at the bottom), then you do not need to be concerned.

When the lights first begin a program in the pattern framework, they should begin by alternating PaleVioletRed and SteelBlue. If they do not, it is likely that the bulbs do not have the right address. The **bulb address** is how we refer to the bulb—so, in the project framework, this is 0, 1, 2, 3, etc. when the bulbs are initially powered on (*i.e.*, plugged in), each expects the first piece of data it receives to be its address and *will interpret any data it receives as an address*. So, if the Arduino board is in the middle of a program when the lights are plugged in, you may end up with weird addressing issues.

To repair these issues, either:

a) Hold down the reset button (it's the red button on the Arduino board) and while holding it down, unplug and re-plug the lights. Once the lights are plugged in again (and the bulbs are thus waiting for an address), release the reset button.

b) Upload the Blink LED program to the Arduino board. This program does not send data to the lights, so the lights are then not receiving any data. Unplug and then re-plug the lights. Upload your program that controls the lights.

Source