**Outline**

Learn the Arduino web based IDE and basic Arduino programming by implementing the basic blink program and modifying it to blink external LEDs.

**Objectives**

* Use constants and variables,
* Explain the difference between syntax, logic, and run-time errors in computer programs;
* Demonstrate the ability to correct syntax, logic, and run-time errors in computer programs;
* Design a simple program from a program template or skeleton (e.g., teacher-supplied skeleton, Help facility code snippet);
* Use Help documentation as a guide to designing and writing programs.
* Use the features of a software development environment to debug programs and create functioning computer programs;
* Work independently, using the Help function, to resolve syntax issues while programming;
* Work independently, using reference materials (e.g., code snippets, sample programs, APIs, tutorials), to design and write functioning computer programs.
* Describe the functions and features of a software development environment and use it to write and run a computer program;

**Materials**

* Arduino Web Development Environment (IDE)

**Level 0: Create an Arduino Account**

On-line Resource: <https://create.arduino.cc/>

NOTE: Always use Chrome for the Web IDE.

1. Go to the website and Select “Arduino Web Editor” and “Create a New Account”. Follow instructions.
2. Check your personal email to confirm your account. Then login and Select “Arduino Web Editor” and agree to terms & conditions (read this first).
3. Note: DO NOT install the plug-in if requested to do so. (Just cancel all dialog boxes.)

**Level 1: Understanding the Blink Example**

1. Follow the editor tour and the tutorial at: <https://create.arduino.cc/projecthub/Arduino_Genuino/getting-started-with-the-arduino-web-editor-4b3e4a>
2. Select the menu item “Examples” 🡪 “Basic” 🡪 “Blink”.
3. Make sure the sketch code is loaded into the editor window.

NOTE: The following steps must be done using the teacher laptop at the front of the classroom.

1. Connect the Arduino board to your PC using the USB cable.
2. Log in to your Arduino Create account.
3. Upload and run the Blink sketch code on the Arduino board.

**Level 2: Using External Documentation**

On-Line Resource: <https://www.arduino.cc/en/Reference/HomePage>

1. Read the on-line documentation to become familiar with the Arduino IDE and command language.
2. Identify each program command and program statement in your blink program.
3. Use the on-line documentation to make notes on each command so you understand and can explain every part of your program.

The setup() function is called when a sketch starts. Use it to initialize variables, pin modes, start using libraries, etc. The setup() function will only run once, after each powerup or reset of the Arduino board.

Loop keeps repeating your sketch over and over again.

Digital write is the option for you to choose either a high or low current.

Pin mode allows you to program specific pins as a input or output device.

Delay is the amount of time you want the program to be paused for.

1. Explain the difference between a constant and a variable.

Constant are variables you do not plan on changing in your Arduino program and they are always the same.

A variable is a place to store a piece of data. It has a name, a value, and a type. They can change based on the circumstances.

1. Explain the following:
   1. Syntax error

A syntax error is when the characters aren’t correct.

* 1. Logic error

A logic error is when there is a error in the code.

* 1. Run-time error

Run time error is when there is an error when the program is running.

**Level 3: Extend the Blink Pattern**

1. Modify the blink program to produce two short blinks followed by one long blink.
2. Use variables to control the blink time.
3. Upload and run the Extended Blink sketch code on the Arduino board and show it to your teacher.

**/\***

**Blink**

**Turns an LED on for one second, then off for one second, repeatedly.**

**Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO**

**it is attached to digital pin 13, on MKR1000 on pin 6. LED\_BUILTIN is set to**

**the correct LED pin independent of which board is used.**

**If you want to know what pin the on-board LED is connected to on your Arduino**

**model, check the Technical Specs of your board at:**

**https://www.arduino.cc/en/Main/Products**

**modified 8 May 2014**

**by Scott Fitzgerald**

**modified 2 Sep 2016**

**by Arturo Guadalupi**

**modified 8 Sep 2016**

**by Colby Newman**

**This example code is in the public domain.**

**http://www.arduino.cc/en/Tutorial/Blink**

**\*/**

**// 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(200); // wait for a second**

**digitalWrite(LED\_BUILTIN, LOW); // turn the LED off by making the voltage LOW**

**delay(200); // wait for a second**

**digitalWrite(LED\_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)**

**delay(200);**

**digitalWrite(LED\_BUILTIN, LOW); // turn the LED off by making the voltage LOW**

**delay(200);**

**digitalWrite(LED\_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)**

**delay(500);**

**digitalWrite(LED\_BUILTIN, LOW); // turn the LED off by making the voltage LOW**

**delay(200);**

**}**

**Level 4: Add External LED**

Useful resources in the “Arduino” folder on the class repository:

* + Getting started guide.pdf
  + Public\_materials🡪Ebook🡪Arduino book.pdf
  + Lessons🡪Lesson1-LED blink

1. Add a second external LED (using components on the prototype board) and make it blink in sync with the on-board LED.

void setup() {

  pinMode(LED\_BUILTIN, OUTPUT);

  pinMode(13, OUTPUT)

}

void loop() {

  digitalWrite(LED\_BUILTIN, HIGH);

  digitalWrite(13, HIGH)

  delay(200);

  digitalWrite(LED\_BUILTIN, LOW);

  digitalWrite(13, LOW)

  delay(200);

  digitalWrite(LED\_BUILTIN, HIGH);

  digitalWrite(13, HIGH)

  delay(200);

  digitalWrite(LED\_BUILTIN, LOW);

  digitalWrite(13, LOW)

  delay(200);

  digitalWrite(LED\_BUILTIN, HIGH);

  digitalWrite(13, HIGH)

  delay(1000);

  digitalWrite(LED\_BUILTIN, LOW);

  digitalWrite(13, LOW)

  delay(200);

}

1. Modify your program so that the external LED blinks following a different pattern than your on-board LED.

void setup() {

  pinMode(LED\_BUILTIN, OUTPUT);

  pinMode(12, OUTPUT);

}

void loop()

{

  digitalWrite(LED\_BUILTIN, HIGH);

  delay(200);

  digitalWrite(12, HIGH);

  digitalWrite(LED\_BUILTIN, LOW);

  delay(100);

  digitalWrite(12, LOW);

  digitalWrite(LED\_BUILTIN, HIGH);

  delay(200);

  digitalWrite(LED\_BUILTIN, LOW);

  digitalWrite(12, HIGH);

  delay(200);

  digitalWrite(LED\_BUILTIN, HIGH);

  digitalWrite(12, LOW);

  delay(1000);

  digitalWrite(LED\_BUILTIN, LOW);

  delay(200);

}

uninstall

1. Upload and run the External LED Blink sketch code on the Arduino board and show it to your teacher.