

NON-ENGINEERING FORMAT FOR HOMEWORK SOLUTIONS

Most problems that require a worked out solution with calculations should be completed using engineering format (GIVEN, REQUIRED, SOLUTION, DISCUSSION) as discussed earlier. For other problems, such as programming your robot or completing a required out-of-class activity, you should use non-engineering format; see the examples below.

Requirements for non-engineering format homework problem solutions:

1. Copy the problem statement (it may be quickest if you do these problems using your word processor).
2. When you are required to write a computer program, always include the program listing as part of your homework; a screen capture works well (using the “snipping tool” in Windows).
3. Write what you have completed and/or learned in your own words. Be concise (don't turn in excessive pages).
4. Problems should be presented in the order in which they are assigned (problem 1 first, problem 2 second, . . .).
5. Number your pages.
6. Remember that it should be easy for a grader to pick up your paper and understand what you have done.

ENGR 120

Homework 2

Solution

1. All students must install SolidWorks to participate in ENGR 120. Click the “helpdesk & events” link on the left sidebar at www.livingwiththelab.com; select “week” on the calendar so you can view the calendar details. Find the software installation date and time for your ENGR 120 section, and go to Bogard Hall 129 on the date listed to install SolidWorks. **You can pay for your software using your Tech Express, VISA, MasterCard, Discover or American Express; checks or cash will not be accepted.** Please see your instructor if you are unable to meet at the designated time for your section.

[I installed SolidWorks on my computer.](#)

2. Refer to the “robot fabrication” presentation available in the “download” section for class 1. If you haven't already done so, please mark the locations of all 10 holes on the aluminum plate so that the Arduino and switches can be attached to the Boe-Bot chassis:

- Four holes to attach the plate to the Boe-Bot chassis
- Two holes to attach the Arduino to the plate
- Four holes to attach the switches to the plate

Bring your parts to class next time ready to punch the holes. You will need to bring your safety glasses. You should complete the assembly of your robot by class 4 (not including the battery pack). Remember that you can work after hours when the HelpDesk is open.

[I plan to assemble my robot during and after class 3.](#)

3. Search the Internet for “Arduino” to locate the Arduino homepage (www.arduino.cc/). Visit the Arduino website and complete the following:

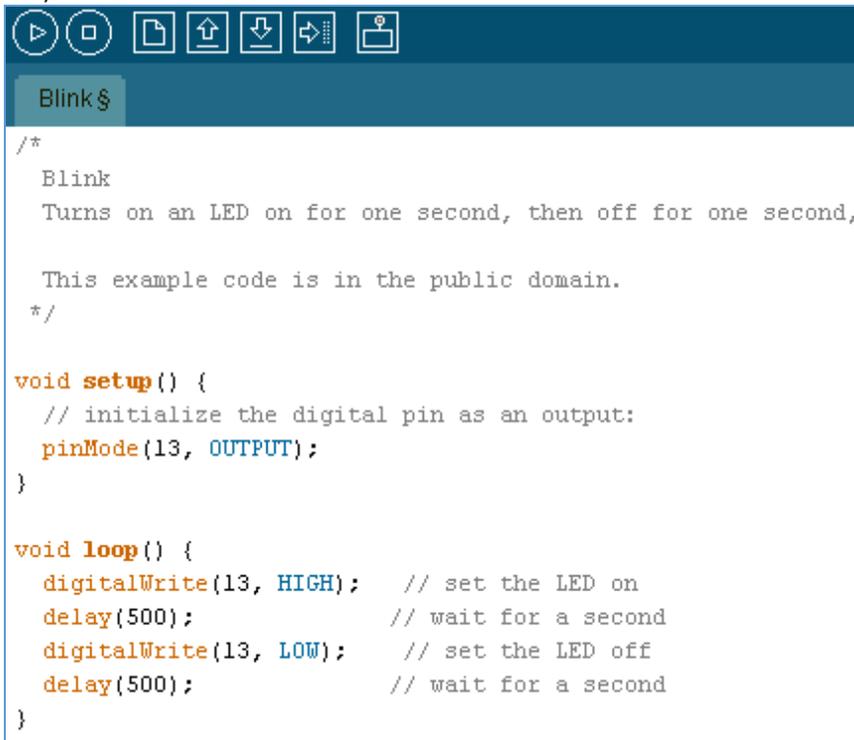
- Click on “Getting Started”
- Click on the appropriate operating system for your computer (Windows, Mac, Linux)
- Follow the directions to . . .
 - Download the Arduino environment
 - Connect the board
 - Install the drivers (if using **Windows 8** and you have trouble, see the NOTE below)
 - Launch the Arduino application
 - Open the blink example
 - Select your board
 - Select your serial port
 - Upload the program
 - Your program should cause the on-board LED to blink

NOTE: The order of the above items depends on your operating system. Regardless of your operating system, you should follow the steps required to make the LED blink. Also, see the downloads for class 2 on the ENGR 120 website for

PowerPoint presentations for Arduino installation on Windows and Mac systems. Slide 20 provides special directions when installing on a Windows 8 operating system.

Answer the following questions for this problem:

- a. Were you able to get the Arduino to communicate with your computer? If not, describe your problems. If you were not, then visit your instructor. If so, then answer questions b through f below. [Yes, I was able to get the Arduino to communicate with my computer.](#)
- b. What happens to the orange LEDs labeled “TX” and “RX” when you upload the program? [They blink really fast . . . this means a program is being transmitted to the microcontroller.](#)
- c. Note that the values of the “delay” commands in the program are “1000”. What happens when you vary these numbers? That is, what happens if you set them to 500? Do you see any changes? [They blink more quickly when you reduce the numbers, and they blink more slowly when you increase them.](#)
- d. What do you think the “delay” command might be doing? [The delay command sets the duration of the blink \(on and off\).](#)
- e. Click on the “language” link under “Reference” on the main Arduino webpage. Click on the “delay()” command and describe what you find (a sentence or two is good). [The delay command pauses the program for the amount of time \(in milliseconds\).](#)
- f. Include a printout of your modified program with your homework (a screen capture would be an easy way to do this).



```
/*
  Blink
  Turns on an LED on for one second, then off for one second,

  This example code is in the public domain.
  */

void setup() {
  // initialize the digital pin as an output:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);  // set the LED on
  delay(500);              // wait for a second
  digitalWrite(13, LOW);   // set the LED off
  delay(500);              // wait for a second
}
```

Solve all of the problems assigned using the appropriate formatting (either engineering or non-engineering format).