Syllabus

CSCI103: Electronic Textiles

Tuesdays & Thursdays; 9:55-11:10am; TCL217


Instructor: Iris Howley

Email: iris@cs.williams.edu

Phone: x4633

Office: TCL308

Office Hours: M 4-5p, T 2-3p, W 3-4p

Overview

Digital data is being infused throughout the entire physical world, escaping the computer monitor and spreading to other devices and appliances, including the human body. Electronic textiles, or eTextiles, is one of the next steps toward making everything interactive and this course aims to introduce learners to the first steps of developing their own wearable technology devices.

After completing a series of introductory eTextiles projects to gain practice in necessary skills, students will propose and design their own eTextiles projects, eventually implementing them with Lilypad Arduino components, and other found electronic components as needed. The scope of the project will depend on the individual’s prior background, but can include everything from a sweatshirt with light-up turn signals for bicycling, to a wall banner that displays the current air quality of the room, to a stuffed animal that plays a tune when the lights go on, to whatever project you can conceivably accomplish with Lilypad Arduino inputs, outputs, and development board in a relatively short time period.

This class is intended for students with no computer programming background, and so a large portion of this class will focus on the development of these skills.

Learning Objectives

By the end of this course, students will be able to:

1. Select appropriate stitch styles and sew two sewable items to each other with that stitch, and secure with knots.
2. Diagram, via paper prototypes, a variety of functioning circuits for physical computing.
3. Implement electric circuit diagrams using electronic components designed for textiles, using a variety of inputs and outputs.
4. Program an Arduino to use a variety of inputs and output components, both analog and digital.
5. Design and create a creative eTextiles product that uses appropriate sewing techniques, simple & complex inputs & outputs, as well as novel Arduino code to transform those inputs & outputs.
6. Apply appropriate debugging techniques that include incremental testing and help seeking from peers, the Internet, the instructor, as well as other resources.

Topics

Course Topics can be found on the Schedule page.

Course Components

This course involves the following components:

- Lectures – we don't have a textbook, so lectures + slides will be our main form of content delivery.
- In-Class Activities and Assignments – individual activities to introduce and practice skills learned in class with immediate feedback. Some will be graded as homework, and some will be participation-based.
- Individual Assignments – it's important to get concentrated hands-on practice with new skills. These individual assignments help you do this.
- Low-stakes Quizzes - will help determine where the class is on certain concepts. They can also be great for helping you self-assess what you need more practice with.
- Projects – larger projects will combine multiple skills into one assignment. These will take more dedicated effort.
- Code Review – formative feedback on your code is useful for being a better programmer.

Textbook

The course will rely heavily on online materials that will be provided as needed. The SparkFun Tutorials on Lilypad will be of particular relevance: learn.sparkfun.com/tutorials/tags/lilypad

Materials

To start, all students will need (provided by instructor, unless otherwise noted):

Available from SparkFun.com

1. 1x Lilypad ProtoSnap Plus Kit (DEV-12922)
2. 1x Lilypad Temperature Sensor (DEV-08777)
3. 1x Lilypad Vibe Board (DEV-11008)
4. 1x Lilypad Tri-color LED (DEV-13735)
5. 1x LilyPad Rainbow LED (6 Colors) (DEV-13903)
6. 1x LilyPad Coin Cell Battery Holder - Switched - 20mm (DEV-13883)
7. 1x Coin Cell Battery Holder – Unswitched – 20mm (DEV-08822)
8. 2x CR2032 Coin Cell Battery (20mm) (PRT-00338)
9. Conductive fabric (COM-14110)
10. Depending on your proposed project, there are other components you may need but are not supplied. Using just the above listed supplies you should be able to implement a variety of interesting projects.

Other

1. Storage box
2. Some fabric to sew on (i.e., craft felt, cotton calico, an old T-shirt, other textiles)
3. Non-conductive sewing thread, embroidery floss
4. Fabric Scissors (Do not use to cut anything other than fabric!!)
5. Needle threaders
6. Depending on your proposed project, you may require specific textile objects that will not be supplied (i.e., Velcro, ribbons, more fabric, a shirt, a stuffed animal, a pennant, painted canvas, etc.). There is a small budget for these supplies, about $5/student.

You may want to download the Arduino Desktop IDE, although the lab machines will have it: [https://www.arduino.cc/en/Main/Software](https://www.arduino.cc/en/Main/Software)

We may use Fritzing to map out our circuits before building them.

**Evaluation**

Strive to do good work because you care about your own opportunities to learn!

There will be graded assignments to provide summative feedback on your progress on course skills. You should complete these assignments individually, without assistance from other students. Class sessions will often include some time to work on these assignments, to provide students timely assistance on their efforts. See assignment hand-outs for specific details.

**Participation and Citizenship**

In order to learn the most from our in-person meetings, it is necessary to attend each session and to complete the assignments. Oftentimes, you will need to show your assignments in class, and so if you have not done the assignment, it will be difficult to discuss your efforts with classmates. This category also includes cleaning up your workspace at the end of class and providing respectful feedback to your peers.

**Grade Breakdown**
Assignments + Quizzes 40%

Class Activities /Citizenship 20%

Final Project 40% (proposal, prototypes, final project)

**Submission**

Many assignments are due the night before class. I will implement this as 11:59pm. In order to be prepared to give you feedback, the instructor must have your submission in the morning. Submitting the day of class, just before class, or in class is therefore unacceptable, risking zero credit.

Each Assignment will describe the Submission process (sometimes we'll use Glow).

**Late work** will be suspect to one letter drop in grade for each day that assignment is late, as well as a 0 for the presentation portion of any assignment with a presentation component.

**Expected Workload**

At Williams, we operate under the course unit system (rather than the credit hour system) as the metric required by many employers, granting organizations, graduate schools, and federal agencies. In addition to the 3 hours we spend together during our class meeting time, you should expect to spend (on average) at least 10 hours per week on the academic and creative work related to class. If you find that you are spending considerably more (or considerably less!) time to engage with this course academically, please contact me so that we can determine the best course of action as you approach the materials. Should you have any additional questions about the relationship of course units to credit hours, refer to the Office of the Registrar, which explains our course unit equivalency in greater detail.

**STUDENTS WHO NEED ACCOMMODATIONS**

If formal accommodations need to be made to meet your specific learning or physical abilities, please contact me as soon as possible to discuss appropriate accommodations. Please also contact the Director of Accessible Education, Dr. G. L. Wallace (413-597-4672) or the Dean’s office (413-597-4171). We will work together to ensure this class is as accessible and inclusive as possible. Also, students experiencing mental or physical health challenges that are significantly affecting their academic work are encouraged to contact me or to speak with a dean. The deans can be reached at 413-597-4171.

**THE HONOR CODE**

Homework and assignments are to be the sole work of each student unless the assignment explicitly states otherwise. Students may discuss issues related to an assignment, provided that such discussions are cited in the material turned in. However, students may not collaborate on
designing or writing code. Uncredited collaborations will be considered a violation of the honor code and will be handled appropriately. For a full description of the Computer Science Honor Code, please see https://csci.williams.edu/the-cs-honor-code-and-computer-usage-policy If in doubt of what is appropriate, do not hesitate to ask.

**Disclaimer**

This syllabus is subject to change at the discretion of the faculty. Students will be notified of such changes ahead of time via email.

**Schedule**

**CSCI103: Electronic Textiles**

**Overview**

Please see the appropriate sub-pages under Assignments for additional direction about Assignments.

**Schedule (Readings, Assignments, Projects)**

**Week 1**

Th 9/5 | Welcome! [slides]

**Week 2**

Tu 9/10 | Websites [slides]

  o READ: Markdown Cheatsheet
  o HANDBOUT: Honor Code Roleplay Scenarios
  o USEFUL:
    * HTML File Paths
    * Beautiful Jekyll Template
  o DUE: Make a github account
  o DUE: Syllabus Quiz

Th 9/12 | Websites [slides]

  o READ: How to Give an Elevator Pitch
  o READ: HTML Cheatsheet
  o USEFUL: PBS Comp Sci Crash Course: The Internet
  o USEFUL (but a lot): How Does the Internet Work?
Week 3

Tu 9/17 | Sewing by Hand [slides]
- USEFUL:
  - 10 Basic Hand Embroidery Stitches for Beginners
- DUE: Project Website Initial Set-up

Th 9/19 | Intro to Programming + Modular Arithmetic [slides]
- READ: What is Modular Arithmetic? on Khan Academy
- USEFUL:
  - C Programming Tutorial on Introduction to C
  - w3resource C Programming Exercises: Declarations & Expressions
- DUE: Practice: Modular Operator on Khan Academy be prepared to discuss in class! (Don't worry about the negative numbers)
- DUE: Modulo Quiz on GLOW

Week 4

Tu 9/24 | Debugging + Conditionals [slides]
- READ the C Programming Tutorial on Introduction to C
- USEFUL:
  - C Programming Tutorial on If Statements
  - w3resource C Programming Exercises: Condition Statement
- DUE: Assignment Embroidery Sampler
- DUE: Deliverable Modular Hello, World!

Th 9/26 | Circuits & Paper Prototyping [slides]
- BRING: Scissors, Pen/Pencil, & Tape (if you have it!)
- USEFUL:
  - Lessons in Electric Circuits: Conductors, insulators, and electron flow
  - Lessons in Electric Circuits: Electric circuits
  - Circuit Simulator
- DUE: Optional Regrade Modular Hello, World
- DUE: Deliverable Sweater Weather

Week 5

Tu 9/31 | Style, Nested-ifs [slides]
- HANDOUT: Parsons' If
- HANDOUT: POGIL Nested If
- DUE: Assignment Simple Circuits (do before the plushy)
DUE: Assignment Circuit Plushy

**Th 10/3 | Conductive Fabric [slides]**

- BRING: All your supplies!
- USEFUL:
  - LilyPad Basics: E-Sewing
- DUE: Plushy Redux
- DUE: Odd Menu

**Week 6**

**Tu 10/8 | While Loops [slides]**

- HANDOUT: Parsons' Nested If
- HANDOUT: POGIL Loops
- USEFUL:
  - C Programming Tutorial on Loops
- DUE: Assignment Homemade Switches

**Th 10/10 | Arduino [slides]**

- BRING: Your Arduino supplies & alligator clips!
- USEFUL:
  - Lilypad USB Plus Hookup Guide
  - Arduino Language Reference Guide
- HANDOUT: Mid-Semester Evaluations
- DUE: Deliverable RGB Loop

**Week 7**

**Tu 10/15 | Reading Period - No Class!**

**Th 10/17 | For Loops + Logic [slides]**

- BRING: Your Arduino supplies & alligator clips!
- USEFUL:
  - C Programming Tutorial on Loops
  - w3resource C Programming Exercises: For Loop
- DUE: Assignment Separate Sparklies x3

**Week 8**

**Mo 10/22 | Iris at the IEEE INFO VIS Conference | Sparklies Lab Session**

- BRING: Your Arduino supplies & alligator clips!
USEFUL:
- Glow > CSCI103 > files > examples > Class-Example-191017
- Glow > CSCI103 > files > examples > Random-Sparklies

Th 10/24 | Iris at the IEEE INFO VIS Conference | Sparklies Lab Session
- BRING: Your Arduino supplies & alligator clips!

Week 9
Tu 10/29 | Arduino Switches, RGB LEDs, Random, Debug Messages [slides]
- BRING: All your Arduino supplies + alligator clips!
- DUE: Assignment 3 Simultaneous Sparklies

Th 10/31 | Light & Temperature Sensors [slides]
- BRING: Your Arduino supplies & alligator clips!
- DUE: Deliverable Random Colors

Week 10
Tu 11/5 | Buzzers [slides]
- BRING: All your supplies!
- USEFUL:
  - Speakers - Magnetism & Sound
  - Physics of Music Notes - Frequencies for equal-tempered scale
- DUE: Deliverable Ghost Detector

Th 11/7 | Functions & Vibe boards [slides]
- BRING: All your supplies!
- HANDOUT: POGIL 12 Void Functions
- HANDOUT: POGIL 13 Value-Returning Functions
- USEFUL:
  - C Programming Tutorial on Functions
  - w3resource C Programming Exercises: Function
- DUE: Deliverable Jammin'

Week 11
Tu 11/12 | Powering & Arrays [slides]
- BRING: All your supplies!
- HANDOUT: POGIL 16 Arrays
READ: Planning a Wearable Electronics Project
REREAD: What are the elements of an effective elevator pitch?
USEFUL:
  - Arduino Language Reference: Arrays
DUE: Deliverable Help, I Need Somebody

Th 11/14 | Project Proposals [slides]
  - BRING: Your project, to work on it!
  - DUE: Project Proposal Document + Pitch

Week 12
Tu 11/19 | Project Workshop [slides]
  - BRING: Your project, to work on it!
  - DUE: Deliverable Analog Arrays
  - DUE: Project Paper Prototypes

Th 11/21 | Project Workshop [slides]
  - BRING: Your project, to work on it!

Week 13
Tu 11/26 | Project Workshop [slides]
  - NOTE: A Photographer will be in class on this day!
  - BRING: Your project, to work on it!

    Thanksgiving Break

Week 14: Project Week
Tu 12/3 | Project Presentations [slides]
  - REREAD: What are the elements of an effective elevator pitch?
  - DUE: Project Reports
  - DUE: Project Demonstration

Th 12/5 | Computer Science & HCI [slides]
  - HANDOUT: SCS Evaluations
Computer Science 103
eTextiles

Iris Howley
iris@cs.williams.edu
Assistant Professor of Computer Science

PLEASE TAKE:
• 1 POGIL 16 Arrays Worksheet
• 1 e-Textiles LiPo battery
• Log-in to your machines
• Log-in to Glow
• Set-up your Arduino

SHARING OUR HELP... DELIVERABLES

Stand-up and state your tip to your past self!

Learning Goals

By the end of this lesson, we should be able to:
• Plug in and unplug a LiPo battery to Arduino
• Define an array & their purpose
• Write code to create new arrays
• Modify existing elements in an array
• Access elements in an array
• Explain why/how insulate thread traces

Powering a LilyPad Arduino

Example:
A project with 10 LilyPad LEDs controlled by a LilyPad Arduino
20mA x 10 = 200mA = 2700mAh (Project mA)

To find out how long a battery will last, use the formula:
Hours = Battery units / Project mA

Let’s use the calculation above to see how long a 110mAh battery will power the project we used in the last example:
0.52 hrs x 110mAh / 2700mA

An e-Textile LiPo battery will only power the project for approximately half an hour!
Powering a LilyPad Arduino

- eTextiles battery lights 10 LEDs for 0.5 hours
- Coin cell battery lights 10 LEDs for 1.2 hours
- You may need to plug your Arduino into a laptop for demoing

You now have eTextiles LiPo batteries
- They are difficult to remove from your Arduino!
- They are rechargeable
  - when Arduino is plugged into computer via USB, with LiPo attached
- If you still need more power, it will be okay to have your project plugged-in to a laptop for demonstrations
  - ~1-2 hours is a long time for these batteries!

Truly made for e-textiles projects.
- PRT-13112
- No risk to short out and a low current delivery
- Will cut off in the range of 240-380mA
- Safest option to power electronics sewn into pieces of fabric.
- Running a few LEDs, you can expect the board to run about 5 hours for every 100mAh of battery capacity.

To recharge an attached battery, plug the board into a USB power source.
- While the battery is charging, the “CHG” LED will illuminate.
- When the battery is fully charged the LED will turn off.
  - Default charge current is set to 100mA
  - 100mAh battery will recharge in one hour, a 1000mAh battery in 10 hours, etc.
  - Do not recommend connecting a lower capacity LiPo battery (i.e. 40mAh LiPo battery) to charge.

Always turn the LilyPad off before inserting or removing a battery.
- Always remove the battery before washing your project and air-dry your project for several days before replacing the battery.
- The battery connector can be tight; to remove a battery never pull on the wires. Use a pair of needle nose pliers or cutters to gently hold-pull the plug out of the connector.

Because the LiPo plugs directly into your Arduino, you do not connect it to any of the sew tabs!
- So you may need/want to rearrange your design if you plan to use it
  - ...also may need to redesign if you’re plugging your Arduino into a laptop for demo day
Powering LilyPad LEDs

- Components need a certain amount of voltage to work
- Longer traces = more voltage drop
  - Keep it short!
- Thinner traces = more voltage drop
  - Use conductive fabric strips

<table>
<thead>
<tr>
<th>Color</th>
<th>Forward Voltage (V)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blue</td>
<td>1.6 volts</td>
</tr>
<tr>
<td>Green</td>
<td>3.2 volts</td>
</tr>
<tr>
<td>Red</td>
<td>2.0 volts</td>
</tr>
<tr>
<td>Yellow</td>
<td>2.0 volts</td>
</tr>
<tr>
<td>White</td>
<td>2.0 volts</td>
</tr>
</tbody>
</table>

Powering LilyPad LEDs

- More components = more voltage drop
  - Spread components across multiple pins
  - Lotsa LEDs? Use more pins!
    - <= 2 LEDs/join for maximum brightness
    - But have seen up to 7 on 1 pin work (not ideal)

- The LiPo battery is preferable, as it is rechargeable.

DON’T WASH:

1. Your LilyPad Buzzer/Speaker
2. Batteries  BAD idea!

Materials

- TBD
- New section on shiny fabrics, cotton, muslin, Velcro, sewable snaps

Materials

- TBD
- Reminder that we can’t use other electronic components
  - Time to ship/arrive
  - Different power requirements
    - Heat pad as example
    - We have so many components! Work with those!
You’ve seen me use a for..loop to turn on a series of LEDs connected to consecutive pins:

```c
for (int ledPin=5; ledPin <=8; ledPin++)
{
    digitalWrite(ledPin, HIGH);
    // delay;digWrite(ledPin,LOW); delay;
}
```

What if the pins aren’t consecutive?

---

**POGIL Activity 16**

- Find a partner and work through the worksheet, Questions 1-4.
  - Done early? Start working on the Application questions.
- All POGIL code is up on Glow
  - Can find the code in: Glow > CSCI103 > Files > examples >
- We’ll come back together and discuss responses as a class afterward.

*Note: I’m short an Arduino & didn’t have one to test this code on. Some may have logic errors!*

---

**POGIL16: Question 1**

- How many elements in `pwms`?
- What type of data is stored?
- Define an array?
POGIL16: Question 2

- What's in `pwms[3]`?
- Declare the last value an OUTPUT?
- Which pin is pin `pwms[2]`?
- What is printed with `Serial.println(pwms)`?
  - How to fix it?

POGIL16: Question 3

- What does `setup()` do?
- What do the 2 `for..loops` do?
- If we add `pwms[4] = 13` to `setup()`, what happens?
- Adjust to use variables, not 4?

arrays – Reassigning Element Values

- Why?
  - Let's say we have 11 LEDs (we do!)
  - And we want to program each to have a different firefly behavior
  - And this behavior has to happen simultaneously
  - We'll want state variables to do this!
  - 11 state variables is a lot
    - Likely easier to keep track of them in an array
    - Need to update the state of the LEDs by reassigning the values of the array elements
    - Also need a boolean array!

POGIL16: Question 4

- Explain `random(0, valsLength)`
- Explain:
  - `int rColor = rgbValues[2];`
  - `int rColor = rgbValues[random(0, valsLength)];`
- What is this code?
  - Random Colors assignment
Arrays & Functions

WE WILL NOT PASS ARRAYS TO FUNCTIONS AS AN ARGUMENT IN THIS CLASS!

(This is a topic that isn’t covered in Computer Science until CS237: Computer Organization)

COMPLETE THE POGIL

• If you didn’t finish the POGIL activities in class, complete them at home!

• Practice with arrays!

Arduino Language Reference Guide


Why insulate conductive thread traces?

• Our conductive thread is not coated/insulated
• So if one thread trace touches another, it forms a new circuit
  • …a short circuit (most likely)
  • Light-up Plushy, anyone?
• To avoid this, we can insulate our traces

Insulating Traces
Tiny patches at crossing point

• Dot of fabric paint (at crossing point)
  With groove to hold the top trace
• Fabric patches (at crossing point)
• Embroidery/satin stitching patch (at crossing point)

Preventing Traces from Touching

Tiny patches at crossing point
**CSCI103: eTextiles**

**Insulating Traces**
- Stretch fabric glue
- Fabric paint

**Insulating Traces**
- Fabric encasement tube
- Embroidery/satin stitching
- Iron-on Interfacing

---

**Homework**

**Project Proposal**
- Project Proposal due 11/13 (Wednesday night)
  - Title, description
  - Materials (be specific about what you need!)
  - Small budget
  - Brief Presentation
- 2 Inputs, 2 Outputs
  - At least 1 analog input, 1 analog output
  - At least one output controlled by input (one of which is analog)
  - Battery Holder switch does not count
- Components must be sewn together with conductive thread

---

**Iris’ Light-Up Iris Bracelet**

Have you ever wanted to wear a conversation starter on your wrist? My light-up Iris bracelet will do just that! When you place a battery in the holder, a pink light will shine through some fancy embroidery.

**SUPPLIES**
- 1 LilyPad LED
- 1 Sewable Battery Holder
- 1 Coin Cell Battery
- Conductive Thread
- Shiny Embroidery floss (pink, yellow, green)
- 6”x12” piece grey Felt
- 1” Sewable Velcro, white
- Thread, Scissors, Needle

---

**Be creative! Be original!**

I will create a simple closed circuit by connecting conductive thread through positive sew tabs on the battery holder and LED using hidden stitch. And another piece of conductive thread through the negative sew tabs (on the battery holder and LED). I will use satin stitch and running stitch to decorate the wristband, and attached Velcro to close it.

---

**SUPPLIES**
- 1 LilyPad LED
- 1 Switched Battery Holder
- 1 Coin Cell Battery
- Conductive Thread
- Shiny Embroidery floss (pink, yellow, green)
- 6”x12” piece grey Felt
- 1” Sewable Velcro, white
- Thread, Scissors, Needle
Making the Pitch Image

1. Powerpoint > File > Export

eTextiles Design Process

- Sketch
  - Layout, Aesthetics, etc.
- Diagram circuit
  - Drawing
  - Paper prototype
- Code the behavior in Arduino
- Test on unsnapped kit
- Alligator Clip Prototype
  - Will have to snap off components!
- Sew!
If at first you don’t succeed...

- Be sure to test everything in small increments!
- When something doesn’t work, can go back to previous stage that does
- If you’re really struggling, try to have something working on demo day
Arduino Activity 16: Arduino Arrays

Name: ___________________ Partner: ___________________

Arduino Activity 16: Arrays

Learning Objectives
Students will be able to:

Content:
- Define an array
- Identify elements of an array
- Explain how to access individual elements of an array

Process:
- Write code that creates an array
- Write code that accesses the elements of an array

Prior Knowledge
- Data types, Expressions, Loops, Random, Arduino, Serial Monitor, analogWrite, RGB LEDs

Critical Thinking Questions:

FYI: Each value stored in an array is called an element.

1. Examine the sample arrays below.

<table>
<thead>
<tr>
<th>Sample Arrays in Arduino</th>
</tr>
</thead>
<tbody>
<tr>
<td>int ledPins[] = {5,6,7,8,12,15,16,17,18,19,20};</td>
</tr>
<tr>
<td>int pwms[] = {6,7,8,10,12};</td>
</tr>
<tr>
<td>int rgbValues[] = {0,128,255};</td>
</tr>
</tbody>
</table>

   a. How many elements does the array named pwms contain? _________________

   b. What type of data is stored in each array?

      • ledPins array: _______________________________________________________

      • pwms array: _______________________________________________________

      • rgbValues array: ___________________________________________________

   c. How would you define an array?

      ______________________________________________________________________

2. This line of code continues from the program in question 1 and declares the first element in the pwms array an output:

   pinMode(pwms[0], OUTPUT);

b. Write a line of code that declares the last value of the array an output.

c. Edit your statement in ‘b’ so that it declares `pwms[2]` an output. What pin is being declared an output? (Try printing it on the Serial Monitor)

**FYI:** The number used to locate an element in the array is called an index.

d. What happens when you try to print the array: `Serial.println(pwms)`? How might we implement what the line intends?

3. Enter and execute the following code, observe the output from your Arduino kit:

```cpp
int pwms[] = {6, 7, 8, 10, 12};
int pwmLength = 5;

void setup() {
  for (int i = 0; i < pwmLength; i++) {
    pinMode(pwms[i], OUTPUT);
  }
}

void loop() {
  for (int i = 0; i < pwmLength; i++) {
    analogWrite(pwms[i], 128);
  }
  delay(500);
  for (int i = 0; i < pwmLength; i++) {
    analogWrite(pwms[i], 0);
  }
  delay(500);
}
```

a. What comment might you write to describe the behavior in the `setup()` function?

b. What comment might you write to describe the first `for .. loop` in the `loop()` function?
c. What comment might you write to describe the second `for..loop` in the `loop()` function?

________________________________________________________________________

d. Add the following line to the beginning of `setup()`, how does the Arduino behavior differ from before? Why?

```cpp
pwms[4] = 13;
```

_______________________________________________________________________
_______________________________________________________________________
_______________________________________________________________________
_______________________________________________________________________

e. How might we adjust the line introduced in d. to use variables, rather than the constant number 4?

_______________________________________________________________________
_______________________________________________________________________
_______________________________________________________________________
_______________________________________________________________________

f. Change the value of `pwmLength` to 3. What happened? Why?

_______________________________________________________________________
_______________________________________________________________________
_______________________________________________________________________
_______________________________________________________________________

4. Enter and execute the following code:

```cpp
int rgbValues[] = {0, 128, 255};
int valsLength = 3;
int rPin = 12; int gPin = 13; int bPin = 14;
int delayTime = 500;

void setup() {
  pinMode(rPin, OUTPUT); pinMode(gPin, OUTPUT); pinMode(bPin, OUTPUT);
  randomSeed(analogRead(0));
}

void loop() {
  int rColor = rgbValues[random(0, valsLength)];
  int gColor = rgbValues[random(0, valsLength)];
  int bColor = rgbValues[random(0, valsLength)];

  analogWrite(rPin, rColor);
  analogWrite(gPin, gColor);
  analogWrite(bPin, bColor);
  delay(delayTime);
}
```

a. Explain the following snippet of code: `random(0, valsLength)`

_______________________________________________________________________

b. Explain the following snippet of code: `int rColor = rgbValues[2];`

_______________________________________________________________________
c. Explain the following line of code:
```cpp
int rColor = rgbValues[random(0, valsLength)];
```

d. How would you describe what this code does?

---

**Application Questions: Use the Arduino IDE to check your work**

1. Create a program that turns a random LED pin to high for a half second, every half second:

2. Revise the previous program so that if the random LED pin it selects is a PWM, it turns that LED on to half the maximum, and keeps it on for one full second.