CodeVA Python SOL   
Final Document

Computer Science SOL Development

Sponsor: CodeVA

CodeVA Project Leads: Bryan Wallace, Rebecca Dovi

**Department: Computer Science**

Faculty Advisor: Dr. Budwell

**Design and Implementation Team:**

#### Kenneth Richardson, Benjamin Napier, Seth Vickers, Dakota Brown, Hunter Frostick

## 

## 

## High-Level Observations of the Capstone Project

The work done by Team 307 revolves around the improvement of the way teachers and students both understand and approach computer science. The tools that we have created, amalgamated, and modified will be used to improve the process by which CodeVA teaches teachers via their Coaches’ Academy.

In regards to the teacher-training aspect of our work with CodeVA, our team has:

* Reviewed CodeVA’s current teacher-training materials (All members)
* Developed new and improved existing Scratch exercises to increase their rigor and effectiveness (3 members)
* Developed Python exercises to mirror the Scratch exercises for teachers and students who are more advanced with text-based languages. (2 members)
* Created presentation slides, derived from the existing Scratch-related slides, to accompany the Python

exercises (2 members)

* Improved upon the current presentation slides to make them more effective, clear, and streamlined (All members)

Additionally, CodeVA has other projects that we worked on in the Spring. These projects were:

* AP Computer Science Principles: create an online AP CSP course and materials for Virtual VA
* Deep Dive Code: create a stand-alone course to introduce teachers to coding fundamentals more deeply
* Dissemination Space for Curriculum: create a digital space to make curricular resources available for teachers to access and download
* Teacher Knowledge Base: help in authenticating and fleshing out CodeVA’s knowledge base of computer science concepts and ideas.
* CityScape: Creating a series of lessons using the client-side library P5.js.
  + This project serves as a stepping stone for teachers and highschoolers looking to initialize or familiarize themselves with a text-based programming language, in this case Javascript.

## Project Completion Details

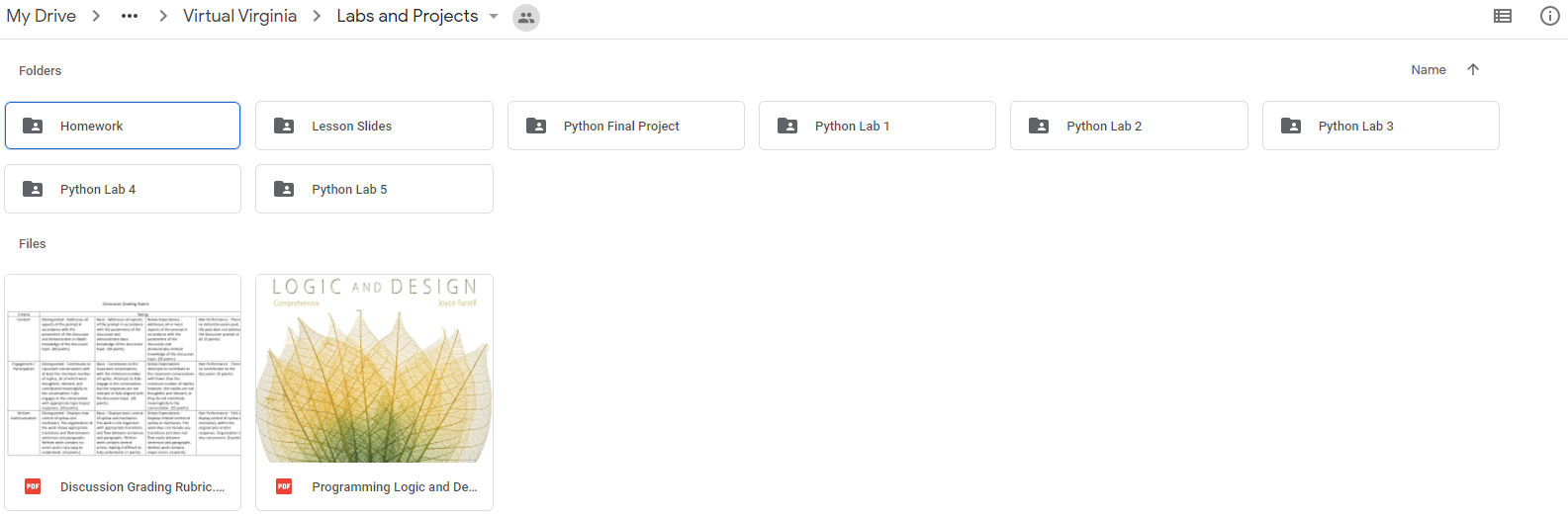
### AP Computer Science Principles Lessons

#### Concept

* + The concept for this project was to create a series of slideshows, labs, and other guidelines for teachers to use in Virtual Virginia’s online AP Computer Science Principles course. These materials serve as an introduction to Python, programming logic, and object-oriented programming. Learning with these lessons should be an interactive process and there are several projects which accompany the lessons as well as one large final project.

#### Final Look

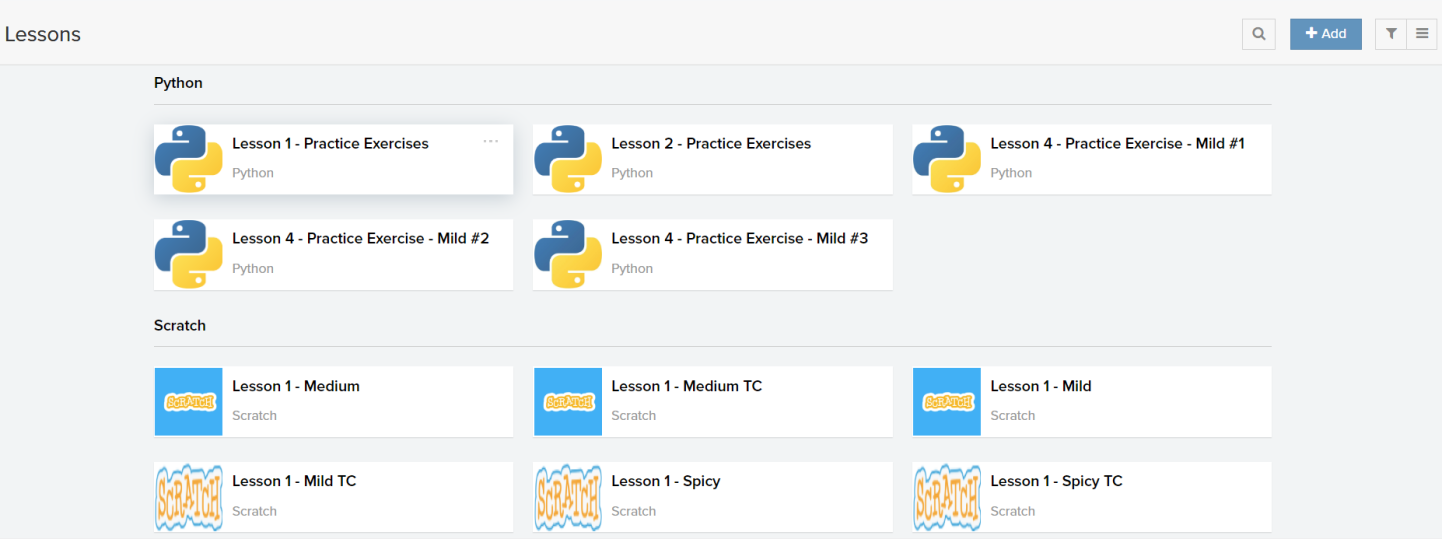
* + Each of these lessons and projects have been either taken from VCU sources or created for this project. Each of the projects and homeworks have keys associated with them to make grading easier for the teachers using this material. There are also suggested rubrics for each project. Links to these resources are below:
    - [Python Lab 1](https://drive.google.com/drive/folders/1vxWAnT5t-sKGKrsUQ896RUXdQd9bEQTf?usp=sharing) - Print statements and running Python for the first time
    - [Python Lab 2](https://drive.google.com/drive/folders/1bSn-swsYI15nbV96nmPkpTPVCi0wLZyn?usp=sharing) - Input/output and calculations using Python
    - [Python Lab 3](https://drive.google.com/open?id=1DHsKtJWBhGOh-h2gYFJDSwwaMmyelBZa) - Python conditionals (if/then, else)
    - [Python Lab 4](https://drive.google.com/drive/folders/1jIGwTgdpp-C_NmCqgLw3NfwP6b9aWqDN?usp=sharing) - More advanced I/O, conditionals for error checking, and variable storage
    - [Python Lab 5](https://drive.google.com/drive/folders/19ef2waKBjSHHjfKDjcXuyLpFzQlkyvJM?usp=sharing) - Code blocks and functions
    - [Python Final Project](https://drive.google.com/drive/folders/1kIK3FByG_lJluq8pNiIxXMJuV_916Qp5?usp=sharing) - Reinforcing all of the previous lab concepts
    - [Lesson Slides](https://drive.google.com/drive/folders/1Hqvah8gjUHfRIzK8hejqgGdZP7wZKDxK?usp=sharing) - These DO NOT teach Python, they teach the fundamentals of programming. Students would be expected to learn Python primarily through the lab exercises.
    - [Homework](https://drive.google.com/open?id=1Z7Y-CmCHYingV3raU6s83ra8y3bj8m2x) - Reinforcing the ideas presented in the lesson slides
    - [Overview of Project Structure](https://drive.google.com/open?id=1DXpXps1OT7KXLSFO66rphDZ3GbKPYg_H) - The structure of the google drive containing this course as a whole
  + The culmination of this project is represented with a Google Drive which contains all of the lesson materials. This material could in the future be included in the CodeVA Dissemination Space or otherwise expanded upon with further time dedicated to the project. This project was on a fairly short time frame due to CodeVA requesting the project to begin in late January so much more could be accomplished with more hours spent.



### Dissemination Space

#### Overview

* + CS educational resources and content openly available to all, not just students and educators
  + Allows anyone seeking to learn CS to download our lesson plans and also to potentially submit their own material to be available on the Dissemination Space.
  + This will make collaborations efforts for better CS learning activities between universities, public schools and communities possible
* Our Lessons Plan page is the user interface the anyone viewing will see. This is where we have added our python and scratch lessons with descriptions and links to download. This is the space where anyone will also be able to add their own lessons that a CodeVa facilitator will review and add if approved.



## 

## 

## 

### CityScape P5.js Lessons

#### Concept

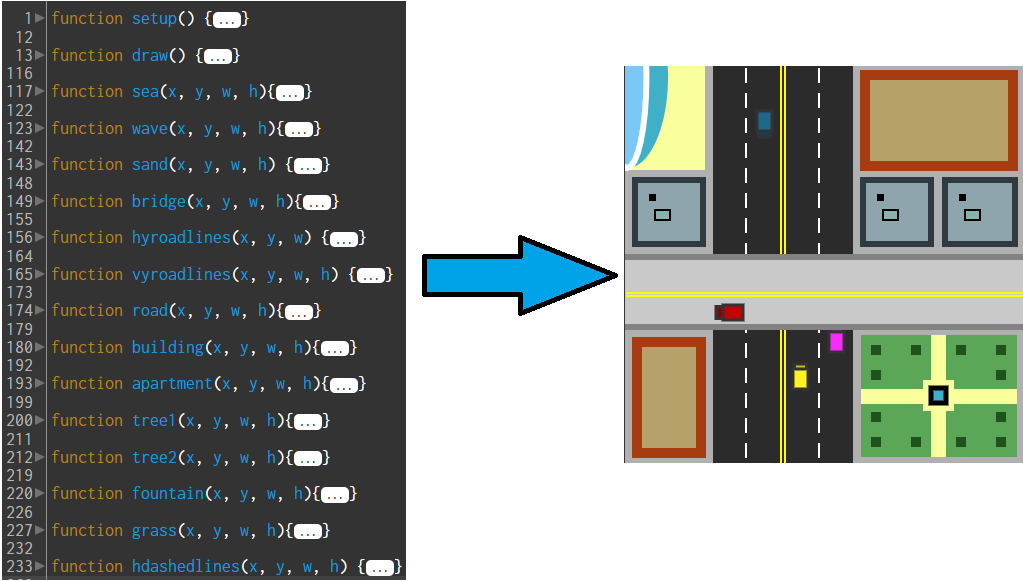
* + The concept for this project was to create a lesson plan for creating a cityscape in p5.js. By doing this, students and teachers are able to see the code we wrote and use the comments as a guideline, walking them through the process of making a simple cityscape while allowing those more familiar with programming the time and directed energy to make something more than what the walkthrough presents.

#### Final Look

* + Here lies the lesson plan we created as well as the links to each lesson

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Lesson Topic** | **Narrative** | **Lesson Link** | **Time** | **url** |
| 0: Introducing p5js | Students are familiarized with the web application of the p5js web editor and the accompanying resources for this tool. | [Lesson 0](https://docs.google.com/document/d/1z57sY1pPChqckrQxWIp6XzEuPbUXT3RSq_E3GUEXKe4/edit?usp=sharing) | 60 min |  |
| 1: The Canvas and Drawing Lines | Covers preparing canvas size and the Cartesian coordinate system for this tool. Students will use lines to connect several (x,y) points after planning on grid paper. | [Lesson 1](https://docs.google.com/document/d/1TU3kivjBkLGrHlEAblTM_qY-mwCuzIYjQBxzkl8_k0w/edit?usp=sharing) | 60 min |  |
| 2: Drawing Shapes | Covers the use of primitive shapes and beginShape(). Students will use these tools to render shapes they have planned out on grid paper. | Lesson 2 | 60 min |  |
| 3: Color & Layering | Covers the use of fill() and the concept of layered rendering. | Lesson 3 | 60 min | <https://editor.p5js.org/alu_templa/sketches/7q3o6J7Q> |
| 4: Variables | Covers the use of variables and their use in setup() as global variables. | Lesson 4 | 30 min | <https://editor.p5js.org/alu_templa/sketches/O4uGK0VU> |
| 5: Objects | Covers the concept of objects. These are unordered, unindexed, and keyed. These should be utilized for complex variables. | Lesson 5 | 30 min | <https://editor.p5js.org/alu_templa/sketches/58xs1jiy> |
| 6: Loops and Variable Updates | Introduces the concept of loops which can be used to draw static objects and updating variables (infinitely). | Lesson 6 | 60 min | <https://editor.p5js.org/alu_templa/sketches/u_X7P8uG> |
| 7: Conditionals | Introduce the concept of conditionals. Very important to stop all of those pesky infinite updates. | Lesson 7 | 60 min | <https://editor.p5js.org/alu_templa/sketches/3Rc2bXjd> |
| 8: Functions | Covers the use of functions for making generic code. | Lesson 8 | 60 min | <https://editor.p5js.org/alu_templa/sketches/a29Kks0g> |
| 9: Wrapper Functions | Introduces the concept of calling a function with a function. | Lesson 9 | 30 min | <https://editor.p5js.org/alu_templa/sketches/r6kHAQMn> |

**Final Product:** [**https://editor.p5js.org/alu\_templa/sketches/oEKRpUEpm**](https://editor.p5js.org/alu_templa/sketches/oEKRpUEpm)



### Deep Dive Code

#### Concept

Create/improve lessons that focus on effortlessly introducing fundamental programming concepts using both block-based and text-based languages, so that transitioning from one to another is seamless. The end goal being a unified set of lessons that could be used to educate teachers on CS fundamentals, in an effort to prepare them for teaching CS SOLs, that they then could use as a guide (to add onto) or as a stand-alone lesson plan for their own classrooms.

#### Final Look

Each lesson introduces and defines a key fundamental computer science concept, using either a

block-based (Scratch) or text-based (Python) programming language to assist. Lessons begin with

definitions and a warm-up exercise (a program) that demonstrates the CS concept being taught.

Learners (whether they be educators themselves or K-12 students) will then gain hands-on experience

via a coding lab in which they are guided through the steps of writing code to demonstrate their

newfound CS conceptual knowledge.

The lessons come in PowerPoint form, with associated coding lab and coding practice exercises being

linked within the slides.

|  |  |  |
| --- | --- | --- |
| **Lesson Name** | **Scratch** | **Python** |
| *Input, Output, and Variables* | [Lesson 1](https://drive.google.com/open?id=1Nlg3ZEttnYV528tCk9htFbssBJhoVVo_) | [Lesson 1](https://drive.google.com/open?id=1kg6yo3PQ6tV8ahWKtOkfJrmg7w4CtwDJ10AflDX_DbA) |
| *Number Calculations and Data Types* | [Lesson 2](https://drive.google.com/open?id=1JBXYamKvNfSa2_uk96cPsBJ30nK8uC7_) | [Lesson 2](https://drive.google.com/open?id=1HDP6E6ZhqZ45W9dS2SStwTnlu0dPcsBB) |
| *Loops* | [Lesson 3](https://drive.google.com/open?id=1VRxG2intR15r0Leeyre8NPz5yBHm5dUH) | [Lesson 3](https://drive.google.com/open?id=1fbfcGngkRQ_uINVS6LX_sxEl_JRvm_g1) |
| *Conditionals - ‘if’ Statements* | [Lesson 4](https://drive.google.com/open?id=1I0rdQ-gCDc-o2w-3klDIFqO2smluBtG7C93d9FrSOl0) | [Lesson 4](https://drive.google.com/open?id=14PYDO4XASaw7KEcsGCVwwlsUyG-g0jEm) |