# Python Ping-Pong Syllabus February 17, 2025

| 1  | Introduction                                     | 1             |
|----|--|---------------|
| 2  | Basic Variables and Print                        | 1             |
| 3  | If Statements<br>3.1 The Boolean Labyrinth       | <b>2</b><br>2 |
| 4  | Calculator Project                               | <b>2</b>      |
| 5  | Python Turtle                                    | 3             |
| 6  | Loops  | 3             |
| 7  | Debugging  | 4             |
| 8  | Advanced Datatypes<br>8.1 Unit Testing for Games | <b>4</b><br>4 |
| 9  | Functions  | 4             |
| 10 | Classes  | <b>5</b>      |
| 11 | <b>Libraries</b><br>11.1 CSV and Data Files      | <b>5</b><br>5 |
| 12 | Exception  | <b>5</b>      |
| 13 | File I/O   | <b>5</b>      |
| 14 | sys Module                                       | 6             |
| 15 | Intro to Pygame<br>15.1 Weather App with Pygame  | <b>6</b><br>6 |
| 16 | Advanced Pygame Techniques                       | 6             |

### 1 Introduction

This course is derived from the <u>The Harvard CS50 Syllabus</u> and study materials are derived from AP Computer Science A. In addition to the skills learned in that course, our Python syllabus prioritizes Python expertise for Pygame Development.

It is highly recommended that you review fractions with your kid as we move along the course.

#### 2 Basic Variables and Print

The student will spend this first class understanding how to create strings and integers and print to the console.

- Strings
- Integers
- Floats (print output trimming)

```
print("Hello, -World!")
print(9)
```

The student will learn to use variables to store data and print multiple variables along with a sentence.

```
a = 5
b = 10
print("a-is:", a, "and-b-is:", b)
```

No floats or booleans yet to avoid confusion. Keep it simple. Teach operators:

```
• +
```

• -

- \*
- \*\*
- / (fallacies of integer division)
- %

### 3 If Statements

- if, elif, and else
- and, or, and not
- <, <=, >, >=, ==, and !=
- Booleans data type and storing conditions in a variable

At this point you can start debugging assignments for certain days and make them a little bit harder too.

Use truth tables to teach advanced concepts like DeMorgan's law. For practice, use the following repos:

- Basic Practice without input
- Program Flow Practice with input
- Advanced Handling (debug skills development)
- Multiple Choice type critical thinking problem set
- Difficult multiple choice/debug problems

### 3.1 The Boolean Labyrinth

- The String Gate: Enter a password that must contain A and end with Z, or be '42'
- $\bullet$  The Integer Trap: Enter a number that's even, not divisible by 5, and  $\natural~10$
- The Boolean Paradox: Enter either True or False. A cat is NOT a dog AND 5 is a string

# 4 Calculator Project

Make a simple calculator that adds, subtracts, multiplies, and divides. You can input() the number, symbol and another number. At the end, the entire equation should be printed. (Example: 1+1=2)

This project tests type casting and basic if usage, which are the more difficult concepts.

At this point it is important to determine what skills need solidification before moving on.

```
5 String Methods
```

- upper()
- lower()
- strip()
- replace()
- split()

Go back and use **split()** for the calculator project.

# 6 Python Turtle

Use graphics to draw simple shapes. Use if statements for complex logic.

- turtle
- goto()
- forward()
- left()
- right()
- done()

Sample assignments:

- Zig-Zag line
- Traffic Light from input

# 7 Loops

- for and while
- in
- break and continue
- range()
- len()
- enumerate()

Also at this point you want to learn:

- match
- Maybe: .sort()

Now begin to draw advanced shapes using Turtle.

- Shape size challenge
- Turtle Race Game where you count steps in a loop

# 8 Debugging

As loops can get buggy, it is important to start debugging now.

- assert
- print
- comments for advanced control flow
- VSCode debugger

# 9 Advanced Datatypes

- Lists
- Dictionaries (store advanced data)

Potentially may return to teach the following concepts although not immediately necessary:

- Tuples
- Sets (no duplicates)

Review the old concepts but up the difficulty of assignments.

#### 9.1 Unit Testing for Games

- Write tests for game mechanics (e.g., assert player.jump() == True)
- Use pytest for automated checks

### 10 Functions

- def
- return
- Return Types
- Call Stack Basics
- Global variables

#### 11 Classes

- class
- Objects
- Methods
- Inheritance
- Initializing a Class

### 12 Libraries

- import
- from
- random
- time.sleep()

12.1 CSV and Data Files

- Save player stats with csv.writer()
- Load high scores with csv.reader()

# 13 Exception

- try, except, and raise
- 14 File I/O
  - open()
  - read()
  - write()
  - close()

At this point, we will start the flagship "code that writes code" assignment.

### 15 sys Module

Introduction to Linux and basic concepts.

- sys.exit()
- sys.argv
- How to run other Python files

Additional libraries likemath may be taught, but aren't necessary for PyGame.

#### 16 Intro to Pygame

???

#### 16.1 Weather App with Pygame

- HTTP requests
- JSON handling
- Rendering

### 17 Advanced Pygame Techniques

- Sprite animation with pygame.image.load()
- Collision detection using rectangles
- Playing sound effects and music