Teaching Outline
Version 3: 3rd March 2016

1. Subject Name: Let's Learn Python

2. Teacher: Aj. Andrew Davison (ad@fivedots.coe.psu.ac.th)

3. Teaching Period: **nine** 2-hour classes; June – July 2016 (??)

4. Coordinator

Aj. Watcharin Kaewapichai (watcharin@coe.psu.ac.th)
Department of Computer Engineering, Faculty of Engineering,
Prince of Songkla University, Hat Yai

5. Course Objectives

To introduce secondary school-level children to basic programming ideas using the Python programming language (version 3.x). Students will be encouraged to write small programs during each class, and run them using the Python IDLE system.

6. Course Description

Topics: installing Python, variables, using modules, if-statements, loops, data structures (lists, tuple, dictionaries), functions. I'll be using the `turtle` module to explain many concepts.

7. Prerequisites

No previous knowledge of programming is required, although the students will need to know how to use a desktop computer.

8. Teaching Method: slide presentations, exercises. Teaching will be done in English. The slides and exercises are written in English.

9. Course Outline

Hour	Topics
1-2	Why Python? Install Python 3.x; IDLE, Python in CMD window; Calculator-style maths, input and print your name Load, run, save a program; Python docs
3-4	What is a variable? (Use 'tag' visuals)

	Get input using input()
	Basic types: integer, float, string.
	Program errors
5-7 (3 hrs)	What is a module?
	Using random, math, turtle, winsound; finding module docs
	Turtle: creating, moving, turning, pen color/width, speed, fill color, coordinates, state. Draw a square, house, triangle, circles, arcs, text
	Downloading extra modules: pyPI, pip
	Installing easygui module; message box, button, choice, text input
8-9	If-statements (using railway line metaphor): else, elif
	Conditionals, and, or, not.
	Exercises 1 handed out.
10-12 (3 hrs)	Loops: counting and conditional
	Turtle examples using counting loops: draw a square, polygon, use of variables & numinput(), spirals (square, angled, circular, adjustable sides, steps)
	Conditional loops examples: times tables, non-termination, continue, break,
	Game examples: number-guessing, ghost game, hangman (self-study)
13-14	Lists: creating, printing, slices, changing, deleting searching
	Tuples
	Grids: list of lists
	Dictionaries: building, accessing, changing, deleting.
	Mutable and immutable data types
15-18 (4 hrs)	Functions: arguments, return
	Local and global variables.
	Examples: silly sentences, Caesar cipher.
	Functions for reusability, clarity, abstraction: draw a smiley
	Simplifying shape drawing; spirograph; random walk; hearts
	Turtles for drawing graphs (self-study)
	Exercises 2 handed out.

10. Assessment

Two programming exercises (worth 50% of the final grade). A 2 hour final exam (worth 50% of the final grade).

11. Teaching Equipment

- White board / black board; a computer, preferably with Internet connection; a projector connected to the computer for showing slides.

- Each student (or small groups of students) should have a computer for doing exercises during the class. I assume the computer will be running a version of Windows, but I could change that if asked.

- In the first class, it would be good if the students could install Python 3.x for themselves (I will be explaining how), but this requires an Internet connection. Alternatively, we can pre-install Python on the student's machines.

12. References

My slides, code examples, exercises, and related information are online at:

```
http://fivedots.coe.psu.ac.th/Software.coe/learnPython/
```

I will include URLs to good sources of information on the Web.

Textbooks

The main book for this subject is:

- **Hello World:** Computer Programming for Kids and Other Beginners
  Warren and Carter Sande
  Manning, 2014, **2nd** ed. (https://www.manning.com/books/hello-world-second-edition)

  It is useful, but not required, reading. Make sure to obtain the second edition which includes more examples.

As far as I know, there is only one Python book written in Thai, but it's intended for university students.

Other Books of Interest

- **Python for Kids**
  Jason R. Briggs
  No Starch Press, 2013, 1st ed. (http://python-for-kids.com/)

- **Python Programming for Teens**
  Kenneth A. Lambert
  Cengage Learning, 2014. 1st ed.
  (http://home.wlu.edu/~lambertk/python/pythonforteens/)
  Despite the name, this is quite advanced.

- **Invent Your Own Computer Games with Python**
  Al Sweigart
  2010, 2nd ed. (http://inventwithpython.com)

  This book can be downloaded from the author's website for free. It's a bit advanced for new programmers. This could be used as the basis for a second course on programming

- **Learn to Program with Minecraft**
  Craig Richardson
  No Starch Press, 2016, 1st ed. (https://www.nostarch.com/programwithminecraft)

  This book uses a Python API for Minecraft to teach programming. It's a nice idea, but complex to set up, and not as interactive as using turtles.

- **Adventures in Minecraft**
  Martin O'Hanlon and David Whale
  Wiley, 2014, 1st ed. (http://www.wiley.com/go/adventuresinminecraft)

  This book uses Python to write Minecraft mods. Aimed at teens. This could be fun as a second course.