**AP Computer Science Principles Assignments**

Welcome to the world of Computer Science. Over the summer I want you to take the time and learn a few things before coming into the class. These assignments are meant to indulge your mind in the beginning of the world of coding.

There will be three different parts of this assignment. Each part plays a very important part in the AP process. Part of advancement in anything is the foundation to understanding. So in essence every part of this assignment is equally important.

**What’s needed for this assignment:**

1. Computer use
2. 3-subject notebook
3. high lighter
4. pen and/ pencil

**Let’s Get Started:**

1. CodeAcademy:
	1. Go to https://www.codecademy.com/learn/python.
	2. Set-up account to keep up with your progress
	3. Go through the entire course and take notes in the first section of your 3-subject notebook.
	4. Included is an example of how your notes should look, use it as a guide.
	5. Print out your completion certificate.
2. Learn Python in one day and learn it Well. (book)
	1. Read the book in its entirety and annotate through your reading. Annotation is when you take small notes in the book or highlight the important information. This is an important skill for English and other subjects. This book will be used, as a guide through the course, so not only is annotation important it is necessary.
3. Final Coding Assignment
	1. Please pick one out of the five of the following projects to write in Python coding. Your final copy of this code should be written in the second part of the 3-subject notebook you need to keep for this class. There should be no scratch out, but a clean copy of your code. Your code should look exactly as it would when you type it in.

**1. Dice Rolling Simulator**

The Goal: Like the title suggests, this project involves writing a program that simulates rolling dice. When the program runs, it will randomly choose a number between 1 and 6. (Or whatever other integer you prefer — the number of sides on the die is up to you.) The program will print what that number is. It should then ask you if you’d like to roll again. For this project, you’ll need to set the min and max number that your dice can produce. For the average die, that means a minimum of 1 and a maximum of 6. You’ll also want a function that randomly grabs a number within that range and prints it.

*Concepts to keep in mind:*

* Random
* Integer
* Print
* While Loops

**2. Guess the Number**

The Goal: Similar to the first project, this project also uses the random module in Python. The program will first randomly generate a number unknown to the user. The user needs to guess what that number is. (In other words, the user needs to be able to *input* information.) If the user’s guess is wrong, the program should return some sort of indication as to how wrong (e.g. The number is too high or too low). If the user guesses correctly, a positive indication should appear. You’ll need functions to check if the user input is an actual number, to see the difference between the inputted number and the randomly generated numbers, and to then compare the numbers.

*Concepts to keep in mind:*

* Random function
* Variables
* Integers
* Input/Output
* Print
* While loops
* If/Else statements

**3. Mad Libs Generator**

The Goal: Inspired by [Summer Son’s Mad Libs project](http://suyeon-son.com/sandbox/mad-libs.html) with Javascript. The program will first prompt the user for a series of inputs a la Mad Libs. For example, a singular noun, an adjective, etc. Then, once all the information has been inputted, the program will take that data and place them into a premade story template. You’ll need prompts for user input, and to then print out the full story at the end with the input included.

*Concepts to keep in mind:*

* Strings
* Variables
* Concatenation
* Print

**4. TextBased Adventure Game**

The Goal: Remember *Adventure*? Well, we’re going to build a more basic version of that. A complete text game, the program will let users move through rooms based on user input and get descriptions of each room. To create this, you’ll need to establish the directions in which the user can move, a way to track how far the user has moved (and therefore which room he/she is in), and to print out a description. You’ll also need to set limits for how far the user can move. In other words, create “walls” around the rooms that tell the user, “You can’t move further in this direction.”

*Concepts to keep in mind:*

* Strings
* Variables
* Input/Output
* If/Else Statements
* Print
* List
* Integers

**5. Hangman**

The Goal: Despite the name, the actual “hangman” part isn’t necessary. The main goal here is to create a sort of “guess the word” game. The user needs to be able to input letter guesses. A limit should also be set on how many guesses they can use. This means you’ll need a way to grab a word to use for guessing. (This can be grabbed from a pre-made list. No need to get too fancy.) You will also need functions to check if the user has actually inputted a single letter, to check if the inputted letter is in the hidden word (and if it is, how many times it appears), to print letters, and a counter variable to limit guesses.

*Concepts to keep in mind:*

* Random
* Variables
* Boolean
* Input and Output
* Integer
* Char
* String
* Length
* Print

**Rubric for summer assignment:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **4****Exemplary** | **3****Accomplished** | **2****Developing** | **1****Beginning** |
| Completion | All requirement are met and exceeded | All requirements are met | One requirement was not completely met. | More than one requirement was not completely met. |
| Organization | Content is well organized an outstanding use of space.  | The uses of space is utilized well.  | Space could have been used better. | Content was not organized. |
| Neatness | All work is exceedingly visible and very precise. | Work is visible and precise. | Work is somewhat visible and a little precise. | Not very visible and not clearly stated. |
| Content  | Displays knowledge beyond intermediate level. | Displays knowledge on an intermediate level. | Displays knowledge on a very basic level. | Doesn’t display knowledge of the content |

**Points Possible: /48**

**CodeAcademy for Python: /16**

**Book annotations: /16**

**Coding Project: /16**