Assignment 20: Final Project Options

Project 1. Hangman

1. Create a file called Hangman.py.
2. It must randomly choose a word from a list of words, stop when all the letters in the word are guessed. It must give them limited tries and stop after they run out. It must display the letters they have already guessed. Suggestion: make a list of the letters in the word.
3. You will be given a list of words to use.

Project 2: Tic Tac Toe

1. You are given starter code in a file called TicTacToe.py.
2. Complete the project so that it makes a 2-player game of Tic Tac Toe, alternatively prompting X and O to choose their location, which is one int representing the row on the board and one int representing the column. The game should end if someone wins (gets 3 in a row) or if the board is full and it is a tie.

Project 3: Password Checker

1. Create a program called PasswordChecker.py
2. The program should prompt the user for a password, then check that it fulfils all the following:
   1. Contains one lowercase letter [a-z]
   2. Contains one number [0-9]
   3. Contains one capital letter [A-Z]
   4. Contains at least one of these: [$#@]
   5. Is at least 8 characters long
   6. Is no more than 16 characters long
3. Print ‘valid’ if this is a valid password and ‘invalid’ if not.

Project 4: Fibonacci Numbers

1. Create a program called Fibonacci.py
2. Prompt the user to enter the number of Fibonacci numbers they want generated. Then, print that many Fibonacci Numbers.
3. The Fibonacci numbers are generated by adding the previous 2 numbers to get the next. The sequence starts with 1 and 1.

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, …

Harder Projects:

**Address Book -** Write an application that acts like an address book. Each entry in it should contain first and last name, phone number, and email. It should be able to sort by at least one field (like last name) - more would be excellent! Next, make it be able to only print entries that match certain criteria (like certain area codes or first names beginning with A).

**Calculator -** Write a program that acts as a calculator. It should take integers and decimals, and do addition, subtraction, multiplication, division, exponents, and give error messages when something is typed wrong. An excellent project will also do modulus, factorial, square roots, and if you press a certain button, it displays a manual of how to use it.

**Quiz Maker** – Make an application which takes various questions form a file, picked randomly, and puts together a quiz for students. Each quiz can be different and then reads a key to grade the quizzes.

**Product Inventory Project** – Create an application which manages an inventory of products. Create a product class which has a price, id, and quantity on hand. Then create an inventory class which keeps track of various products and can sum up the inventory value.

**Movie Store** – Manage video rentals and controls when videos are checked out, due to return, overdue fees and for added complexity create a summary of those accounts which are overdue for contact.

**Student Grade Book Application** – Keep track of students (with a student class that has their name, average, and scores) in a class and their grades. Assign their scores on tests and assignments to the students and figure out their average and grade for the class. For added complexity put the students on a bell curve.

**Bank Account Manager** - Create a class called “Account” which will be an abstract class for three other classes called “CheckingAccount”, “SavingsAccount” and “BusinessAccount”. Manage credits and debits from these accounts through an ATM style program.

**Library Catalog** – Create a book class with a title, page count, ISBN and whether or not it is checked out or not. Manage a collection of various books and allow the user to check out books or return books. For added complexity generate a report of those books overdue and any fees. Also allow users to put books on reserve.