APCSP

Unit 2 – Developing Programming Tools

Project

Directions: Choose 1 program from each category. Write the program in python using your choice of IDE. Include comments at the start of your program for your name, problem number, program description as well as comments throughout to indicate other contributors and notes of program components. Your program also must incorporate several functions to break apart and re-use parts of code. Programs include a way to earn optional "bonus" points, but these can only be earned if each of the required programs have been successfully completed. On a google doc copy the code and the output. After each completed program write a couple of sentences to describe difficulties and/or opportunities encountered AND how you resolved or incorporated them.

Rubric: This rubric will be used throughout the grading of each program. (total of 12 points each)

Remarks		**Functions**			**Efficiency**			**Structures**			**Readability**			**Output**		**Reflection**		
All required remarks in program		Functions used in a manner to improve both readability and efficiency			Efficiency evident through structures, functions and variable type (lists, etc.)			Appropriate coding structures such as loops, conditionals used where needed			Spacing, variable choices and function name increase readability throughout the program.			Output of various types of input are shown in final google doc		Multiple sentence reflection is written with both opportunities and resolutions.		
0	1	0	1	2	0	1	2	0	1	2	0	1	2	0	1	0	1	2

Category 1: Strings

Program A: Hangman

You are to write a hangman program in which one player inputs the key word (to be guessed). Display the blanks to be filled and some representation of a "hangman." The other player can then guess letters until either the word is guessed (they win) or the predetermined number of guesses has been reached (they lose). Bonus points could be awarded if the program already had some words built in and it selected the word, making this essentially a one person game, not two.

Program B: Pig Latin

You are to write a program that allows a user to enter a word and translate it into the language of Pig Latin. Here are the rules:

1. If a word starts with a consonant and a vowel, put the first letter of the word at the end of the word and add "ay." Example: happy = appy + h + ay = appyhay
2. If a word starts with two consonants move the two consonants to the end of the word and add "ay." Example: child = ild + ch+ay = ildchay
3. If a word starts with a vowel add the word "way" at the end of the word. Example: awesome = awesome + way = awesomeway

Bonus points could be awarded if the program can take an entire sentence or phrase and translate every word as well as one word at a time.

Program C: Palindrome

A palindrome is a string that is the same forwards and backwards (i.e. Hannah, kayak and radar). Your program should allow a user to input a string and your program should determine if the string is or is not a palindrome. Bonus points may be awarded if your program also correctly identifies palindromes with spaces like race car and taco cat.

Category 2: Loops within a random number application

Program A: "Run For It" dice game

Your program should simulate a game of "Run For It" dice game. In this game players roll 6 dice and if their set of dice includes a sequence starting with 1 like 1-2-3-4, they are awarded 5 points for each number in the

sequence. For example, a roll of 3,2,5,4,4,1 has a sequence of 1-2-3-4-5 and the player is awarded 25 points; a roll of 3,2,4,5,3,6 has no sequence beginning with 1 and the player is awarded 0 points. The game continues until a player has 100 points. Bonus points may be awarded if your program allows for 2-6 players rather than 1 or 2.

Program B: Rock, Paper, Scissors

Your program should simulate a game of "Rock, Paper, Scissors" with two players. Make sure if players tie then they get to go again. The winner should win 2 out of 3 games. Bonus points may be awarded for appropriate graphics accompanying the game.

Program C: Guess the number

Your program should allow one user to input a number, then prompt another user to guess that number. Prompt the second user to continue to guess until they have the correct number. Include suggestions such as "too high" or "lower" to help direct them to the correct answer. Bonus points may be awarded if the program allows for a one player game meaning that the program generates the number to guess.

Category 3: Mathematical Algorithms

Program A: Coordinate Geometry with Reflections and Rotations

This program should allow a user to enter a set of coordinate points (at least 3 up to 6). Your program should then perform various reflections (over x-axis, over y-axis, over y = x and over y = -x) and rotations (90°, 180°, and 270°). Your program should output the new points over each reflection and rotation. Bonus points may be awarded for graphics – those graphics are probably best done in turtle graphics.

Program B: Rectangle

This program will do several operations on a rectangle. First, allow the user to enter a set of coordinate points that would represent the vertices of the rectangle. Your program must verify that it is a rectangle (based on slopes and/or side lengths) then calculate and output the perimeter and area of the rectangle. Bonus points may be awarded for a graphical representation of the shape using turtle graphics or additional calculations with the rectangle such as the equation for the lines that for the diagonals.

Program C: Solve a quadratic with the quadratic formula

This program should calculate the solution(s) of a quadratic equation using the quadratic formula. The user should be prompted to enter the values for a, b and c ($ax^2 + bx + c = 0$). Your program should output the value(s) for x. Bonus points may be awarded for a correct graphical representation perhaps using turtle graphics and/or for various methods of output including the original equation where your program isolates the a, b and c.