**COMP130 HW7: Iteration
instructor: John MacCormick**

Question 1. (5 points) Rearrange the program statements below into a program that displays all of the multiples of 5 between 0 and 50 inclusive. Make sure to use correct indentation.

num = num + 5

print(num)

while num <= 50:

num=0

Question 2. (5 points) Consider the program below:

x=3

while(x != 100):

 print(x)

 x = x + 2

Explain why the while loop in this program causes an infinite loop.

Question 3. (10 points) Rewrite the program below so that it generates the same output but uses a while loop instead of a for loop.

line = '\*'

for level in range(10):

 print(line)

 line = line + '\*'

Question 4. (5 points) Consider the following program.

import random

num=0

tries=0

while num != 50:

 num=random.randint(1,100)

 tries = tries + 1

print('It took ' + str(tries) + ' tries to get 50.')

In your opinion, does this program cause an infinite loop? Explain your answer in one sentence.

Question 5. (5 points) Rearrange the statements below (adding indentation) to produce a program that has the same behavior as the one in the previous question.

print('It took ' + str(tries) + ' tries to get 50.')

import random

break

tries = tries + 1

while True:

if (num == 50):

tries=0

num = random.randint(1,100)

Question 6. (10 points) Rewrite the following program so that it accomplishes the same task, but does not use any break statements.

from graphics import \*

"""

Draw circles at each point where the mouse is

clicked until Q or q is pressed, then exit the

program.

"""

win=GraphWin('CirclePad',500,500)

while True:

 mouse=win.checkMouse()

 if mouse != None:

 cir=Circle(mouse,20)

 cir.draw(win)

 key=win.checkKey()

 if key=='Q':

 break

 elif key=='q':

 break

win.close()

Question 6. (60 points)

The task in this question is to implement the dice game Pig for two players. Our version of Pig will have one dice and two players. The players take turns. On each turn, the current player rolls a dice. If the roll is 2, 3, 4, 5, or 6, then the dice value is added to the player's *turn score*. The player can choose to *stop* and add their turn score to their total score; or the player can choose to roll again, hoping for a larger turn score. If the roll is a 1, then the player's turn ends and their turn score is 0. The player continues until either they choose to stop or they roll a 1. Then the other player takes their turn. The first player to accumulate a total score of 100 wins. A [YouTube video explaining how to play Pig](https://www.youtube.com/watch?v=pJZsQ5528ZM) is available.

At the end of this assignment, some skeleton code is provided. Your task is to complete this program so that:

* The players take turns.
* On each turn, the player can roll as many times as they like or until they roll a 1.
* A turn score and total score is maintained for each player
* The game continues until one of the players reaches 100 points.
* The winner of the game is displayed.

In addition, the program:

* Should be tolerant of incorrect choices and ask again if an invalid value is entered by the user.
* Should reflect good design (readability, encapsulation, generalization, clean interfaces).

The output of a working program might be similar to the following:

Player 1 rolls a 4

Player 1 turn score 4, total score 0

[R]oll or [S]top: R

Player 1 rolls a 2

Player 1 turn score 6, total score 0

[R]oll or [S]top: T

T is not an option.

[R]oll or [S]top: R

Player 1 rolls a 6

Player 1 turn score 12, total score 0

[R]oll or [S]top: S

Player 1 total score: 12

Player 2 rolls a 3

Player 2 turn score 3, total score 0

[R]oll or [S]top: R

Player 2 rolls a 1 - Bust!

Player 2 total score: 0

Player 1 rolls a 5

Player 1 turn score 5, total score 12

[R]oll or [S]top: R

Player 1 rolls a 2

Player 1 turn score 7, total score 12

[R]oll or [S]top: ...

...

Player 2 rolls a 3

Player 2 turn score 3, total score 95

[R]oll or [S]top: R

Player 2 rolls a 4

Player 2 turn score 4, total score 95

Player 2 total score: 101

Game Over!

Player 2 wins!

Here is the skeleton code for Pig:

import random

"""

A program that plays a 2-player version of the dice game Pig.

"""

print("Let's play Pig!")

player1\_score = 0

player2\_score = 0

win\_score = 100

while (player1\_score < win\_score) and (player2\_score < win\_score):

 # Run player 1's turn

 player1\_score = player1\_score + 10 # replace with actual turn total later.

 # Run player 2's turn

 player2\_score = player2\_score + 12 # replace with actual turn total later.

print("Game over!")

# Print the winner.

Total points on assignment: 100