# Coursework Workshop – 31/10/2022

Approach:

* Understand the problem.
* Create the flowchart / pseudocode.
* Translate the flowchart / pseudocode into Python Code.

## Part 1 - Main Version

1. Outcomes (28 marks)

* The program should allow students to predict their progression outcome at the end of each academic year. The program should prompt for the number of credits at pass, defer and fail and then display the appropriate progression outcome for an individual student (i.e., progress, trailing, module retriever or exclude).

What we have covered:

* input()
* if/elif/else

possible pseudocode.

GET pass credits

GET defer credits

GET fail credits

If pass is equal to 120

Display “Progress”

ENDIF

(add additional conditions for trailer, retriever, exclude – could be 3 or 27 extra conditions)

Validation (12 marks)

* The program should display ‘Integer required’ if a credit input is the wrong data type.

#### # Loops - Seminar

#### # Exercise 3 - Revision of exception handling.

while True:

    n = input("Please enter an integer: ")

    try:

        n = int(n)

        break # break or continue

    except ValueError:

        print("Requires a valid integer! Please try again.")

print("You successfully entered an integer.")

* The program should display ‘Out of range’ if credits entered are not in the range 0, 20, 40, 60, 80, 100 and 120.

You have data available to you - Inputs for pass, defer and fail.

You can make a check that these meet the criteria. Many ways to do this – students should select their approach.

* The program should display ‘Total incorrect’ if the total of the pass, defer and fail credits is not 120. If

GET pass credits

GET defer credits

GET fail credits

CALCULATE total of pass, defer and fail

If total is equal to 120

If pass is equal to 120

Display “Progress”

Endif

(add additional conditions for trailer, retriever, exclude)

Else

Display ‘Total incorrect’

Endif

* A few marks will be allocated for the efficient use of conditional statements. For example, the program does not need 28 conditional statements for 28 outcomes.
* An example of the program running with user input (shown in bold):

Please enter your credits at pass: p

Integer required

Please enter your credits at pass: 140

Out of range.

Please enter your credits at pass: 100

Please enter your credit at defer: 40

Please enter your credit at fail: 20

Total incorrect.

1. Multiple Outcomes (12 marks)

* The program loops to allow a staff member to predict progression outcomes for multiple students.
* The program should prompt for credits at pass, defer and fail and display the appropriate progression for each individual student until the staff member enters ‘q’ to quit. Optionally you can use an input of ‘y’ to continue.

Loop - which loop is best (while or for loop) and why?

For suitable while loops, check out the Blackboard area:

‘Example Programs (some seminar solution)’.

For example: while True or using a sentinel value to control the while loopBottom of Form

possible pseudocode.

SET progress count to zero (for how many progress (for histogram))

SET total inputs to zero (for how many outcomes we input)

WHILE True

INPUT pass credit

INPUT defer credit

INPUT fail credit

CALCULATE total of pass, defer and fail

If total is equal to 120

Add one to total inputs

If pass is equal to 120

Display “Progress”

Add one to progress count

Endif

Else

Display ‘Total incorrect’

Endif

INPUT more entries

If more entries is equal to ‘q’

Exit WHILE

ENDWHILE

DISPLAY histogram

1. Histogram (8 marks)

* When ‘q’ is entered, the program should produce a ‘histogram’ where each star represents a student who achieved a progress outcome in the category range: progress, trailing, module retriever and exclude. The histogram should relate to the data input entered during the program run and work for any number of outcomes.
* Display the number of students for each progression category and the total number of students.
* Example of a program run and input (in bold). Note: program should exit on ‘q’ to quit. ‘y’ to continue shown in the example is optional and depends on your program structure.

Example Output:

Enter your total PASS credits: 120

Enter your total DEFER credits: 0

Enter your total FAIL credits: 0

Progress

Would you like to enter another set of data?

Enter 'y' for yes or 'q' to quit and view results: y

Enter your total PASS credits: 100

Enter your total DEFER credits: 0

Enter your total FAIL credits: 20

Progress (module trailer)

Would you like to enter another set of data?

Enter 'y' for yes or 'q' to quit and view results: y

Enter your total PASS credits: 80

Enter your total DEFER credits: 20

Enter your total FAIL credits: 20

Module retriever

Would you like to enter another set of data?

Enter 'y' for yes or 'q' to quit and view results: y

Enter your total PASS credits: 60

Enter your total DEFER credits: 0

Enter your total FAIL credits: 60

Module retriever

Would you like to enter another set of data?

Enter 'y' for yes or 'q' to quit and view results: y

Enter your total PASS credits: 40

Enter your total DEFER credits: 0

Enter your total FAIL credits: 80

Exclude

Would you like to enter another set of data?

Enter 'y' for yes or 'q' to quit and view results: q

---------------------------------------------------------------

Histogram

Progress 1 : \*

Trailer 1 : \*

Retriever 2 : \*\*

Excluded 1 : \*

5 outcomes in total.

----------------------------------------------------------------

* Submit the completed part 1 test plan provided with your final part 1 solution.

#Loops seminar: Exercise 6 (displaying stars)

# Option 1

stars = int(input('How many stars? '))

for x in range(stars):

   print('\*', end='')

print()

#Option 2

stars = int(input("How many stars? "))

print('\*' \* stars)

However, the values will be stored in variables that you update as the program runs.