

0250002

by Iskander Abd-al-kerim

FILE	18864_ISKANDER_ABD-AL-KERIM_0250002_341067_1080541608.DOCX (104.64K)		
TIME SUBMITTED	26-APR-2021 01:47PM (UTC+0100)	WORD COUNT	4097
SUBMISSION ID	150536953	CHARACTER COUNT	21436

Introduction to Computer Programming
ACCA4020 Programming in Python

Lecturer G Thomas

Iskander Abd-Al-Kerim
0250002 @student.uwtsd.ac.uk

1. Abstract

As is the course it is part of this report aims to present the reader with an introduction into computer programing. It particularly focuses of python as being the programing language used in the questions it answers and the examples it presents. It will ask question and show examples relating to variables, statements, and loops.

Then some exercises are tackled, solutions are presented for writing programs that execute certain functions and a basic insight is offered into computer programing.

Contents

1. Abstract.....	1
2. Introduction.....	2
3. Main.....	3
3.1. What is a Variable & what importance do they hold in programming?	3
3.2. What are if/else statements & how are they used?	4
3.3. Why are loops needed in programs and when would they be used?	5
4. Exercises	6
4.1. Exercise 1	7
4.2. Exercise 2	8
4.3. Exercise 3	10
4.4. Exercise 4	11
References	18

2. Introduction

These days most people do not need to be introduced to computers and how they work. However, when delving into computer programming a little more is required than how to simply turn a computer on and off. [1]

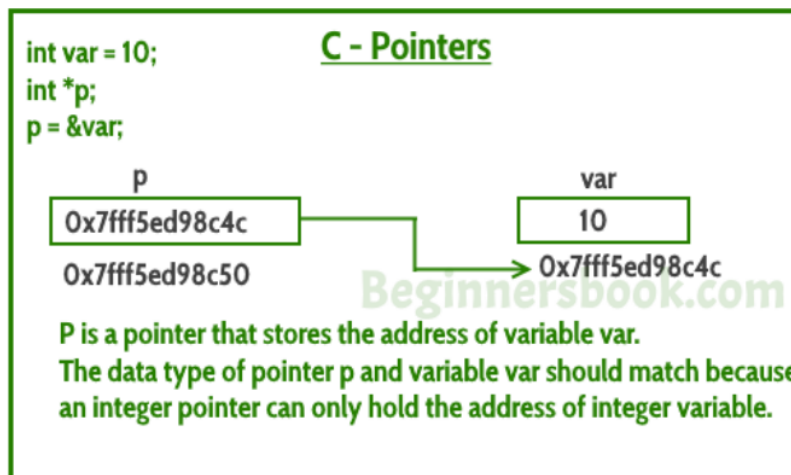
In this document, the reader will be introduced to various aspects of Computer Programming particularly those relating to the programming language known as Python.

3. Main

Looking at three very serious questions relating to computer programming.

3.1. What is a Variable & what importance do they hold in programming?

A variable is a name for an object stored in memory. Usually written in hexadecimal, a name for a variable memory address location is known as a pointer. [2]



[3]

Fundamentally all programming boils down to variables.

```
x = 5
y = "John"
print(x)
print(y) [4]
```

Selecting a type for a variable is like selecting what action can be taken concerning said variable.

```
x = 4 # x is of type int
x = "Sally" # x is now of type str
print(x) [4]
```

Particularly in Python, everything you do or interact with can be classed as a type of object such as classes, modules, sets, functions, tuples, files, strings, lists, dictionaries, numbers. [5]

Type casting for example will allow for variable value type to be set:

```
x = str(3)    # x will be '3'
y = int(3)    # y will be 3
z = float(3)  # z will be 3.0 [4]
```

The importance for variables in computer programming comes from how they enable programmers to write flexible programs. Its an alternative to entering directly into a program this allows a programmer to use a variable as a representation for the data.

```
a = 4
A = "Sally"

print(a)
print(A)
```

Then prints,

```
4
Sally [6]
```

At the point the program is then executed the variable is executed with the real data. This in turn enables the program to process varying sets of data. [7]

3.2. What are if/else statements & how are they used?

By using the “if” keyword you can start a “if statement” which can then lead into a “elif” stating else if and finally finishing of with an “else” the “elif” is only used if there are three or more conditions withing the statement to be tested by the program.

The logical mathematical conditions may be: [8]

1. Equals: $a == b$
2. Not Equals: $a != b$
3. Less than: $a < b$
4. Less than or equal to: $a <= b$
5. Greater than: $a > b$
6. Greater than or equal to: $a >= b$

Looking at how to write a if statement with three conditions to show Equals, A is more than B and A is less than B.

```
# we will create two variables and get their value from user input then we
will test if they are equal, A is
# greater than B or if A is less than B. to top it off there is a function
added to ensure only a interger is inputed

def input_number(message): # inputNumber ensures user input a numerical
value and in the event another value is
# entered it prompts for correct input
    while True:
        try:
            userInput = int(input(message))
        except ValueError:
            print("Not an integer! Try again.")
            continue
        else:
            return userInput
            break

A = input_number("Enter whole numerical value for A: ")
B = input_number("Great now enter a whole numerical value for B: ")

if A == B:
    print("The two numbers a re equals")
elif A < B:
    print("A is less than B")
else:
    print("A is greater than B")
```

This program allows and individual to make use of if/else statements to determine the difference between two numerical values.

3.3. Why are loops needed in programs and when would they be used?

Loops are a repetitive control structure in programming languages [9]. A great example of a for loop being used can be found at rosettacode.org [10]. For this example, the for loop was placed withing a function to allow it to be called as and when it is required by the programmer.

```
# the following code shows anexample of how a for loop could be used for
repetitive text then it is added to a
# function so it is easily called upong whenever it becomes necessary within
the code.

def song_numbers():
    for i in range(2, 9, 2):
        print("%d, " % i, end="")

print(song_numbers(), "who do we appreciate?")
print(song_numbers(), "who do we appreciate?")
```

Which would in turn print the following:

Iskander Abd-Al-Kerim - 0250002@student.uwtsd.ac.uk

2, 4, 6, 8, None who do we appreciate?
2, 4, 6, 8, None who do we appreciate?

As can be seen the programmer no longer has to repeat writing the text “2, 4, 6, 8,” the for loop does this for them. This sees the program being told to start counting at 2 and to finish at 8 as 9 is second in the brackets it will stop count at 1 before second number. The third number then tells the program that it is required to count incrementally rather than by the default of one it is required to go up in 2s. So start at 2, finish at 1 before 9 and count in 2s. [11]

Withing the answer to Exercise 4 there can be seen a decent example of when and how a while loop can be used and why:

```
while user_w_width < 15 or user_w_width > 45: # here a while loop is used to ensure that the correct
# number is inputted
    user_w_width = input_number("Try again and this time input a whole number between 15 and 45: \n")
print(hero_hard.penup(), hero_hard.back(word_width * 15), hero_hard.pendown(), turtle.clearscreen(),
      pick_word())
```

Here the while loop is used to ensure the user input is no greater than 45 and no less than 15 which allows the creates a confirmation in a sense that the integer inputted by the user is with the desired parameter in order to allow them to progress further into the program.

4. Exercises

4.1. Exercise 1

```
# Exercise 1 allowing user to input details relating to a car then
displaying them

def doors(): # this is our doors function it will allow the user to input
anything from 1 - 5 doors and then print
    # all inputs else it will ask user for another input for doors
    print("Please enter the number of doors this can be anything from 1 -
5")
    door_count = int(input()) # user inputs doors on car
    if door_count == 1:
        print("Ok so were looking at a", car_make, car_model, "with",
round(mileage, 1), "miles on the clock and",
        door_count, "doors.")
    elif door_count == 2:
        print("Ok so were looking at a", car_make, car_model, "with",
round(mileage, 1), "miles on the clock and",
        door_count, "doors.")
    elif door_count == 3:
        print("Ok so were looking at a", car_make, car_model, "with",
round(mileage, 1), "miles on the clock and",
        door_count, "doors.")
    elif door_count == 4:
        print("Ok so were looking at a", car_make, car_model, "with",
round(mileage, 1), "miles on the clock and",
        door_count, "doors.")
    elif door_count == 5:
        print("Ok so were looking at a", car_make, car_model, "with",
round(mileage, 1), "miles on the clock and",
        door_count, "doors.")
    else:
        print(doors())

print("Please enter a car make: ") # this is actually the start of the
program where a car make is imputed
# this can be in any foprmat or length
car_make = input()

print("Please enter a car model: ")
car_model = input() # user inputs car model

print("Please enter the mileage. Input must be 0 miles upwards")
mileage = float(input()) # user inputs car mileage

while mileage < 0: # this while loop checks to ensure the a negative milage
count was not inputed and if it was
    # repromts for a positive input to be made
    print("Please enter the mileage. Input must be 0 miles upwards")
    mileage = float(input()) # user inputs car mileage
else:
    print(doors()) # once a corecting iput for milage is aquired we then
call the door function
```

```

C:\Users\iaak1\AppData\Local\Programs\Python\Python39\python.exe "C:/Users/iaak1/Desktop/Leam
Python/assex1.py"
Please enter a car make:
Honda
Please enter a car model:
Acord
Please enter the mileage. Input must be 0 miles upwards
-5
Please enter the mileage. Input must be 0 miles upwards
100
Please enter the number of doors this can be anything from 1 - 5
8
Please enter the number of doors this can be anything from 1 - 5
3
Ok so were looking at a Honda Acord with 100.0 miles on the clock and 3 doors.
None
None

```

Process finished with exit code 0

4.2. Exercise 2

```

# Exercise 2 enter shape dimensions and calculate results. It also informs
the user if the area covered by the
# triangle is greater than that of the circle there are also while loops
being use to ensure each input reflects
# a positive amount

print("Enter the radius of a circle then enter the base and height of a
triangle and we will tell your what is"
      " left in the circle area if we deduct the area of the triangle.
Please be sure you enter values in CM")
pi = 3.142 # we use pi variable
print("Please enter the radius")
radius = float(input()) # user inputs radius
while radius < 0: # while loop ensures a positive amount is being entered
    print("Please enter the radius and be sure you input a positive
number")
    radius = float(input()) # user inputs radius

circle area = (radius ** 2) * pi # radius to power of 2 times pi

print("Please enter the height")
height = float(input()) # user inputs height
while height < 0: # while loop ensures a positive amount is being entered
    print("Please enter the height and please be sure you input a positive
amount")
    height = float(input()) # user inputs height

print("Please enter the base")
base = float(input()) # user inputs base
tri area = (height * base) / 2 # triangle base times height divided by 2 #
gives us triangle area
while base < 0: # while loop ensures a positive amount is being entered
    print("Please enter the base and ensure you are entering a positive
number")
    base = float(input()) # user inputs base
    tri_area = (height * base) / 2

```

```

area_left = circle_area - tri_area # we deduct triangle area from circle
if area_left < 0: # if circle cover less space than triangle we get a
answer below
    # zero so we tell the user how much smaller the circle is
    print("It seems the area of your circle is", area_left, "cm too
small for your triangle")
else: # otherwise we tell them how much is left over opnce the triangle
area
    # is deducted
    print("The total area left is", area_left, "cm.")

```

Following two outputs both show the code responding to negative number being inputted the first example the result when the area of the triangle is greater than that of the circle

C:\Users\iaak1\AppData\Local\Programs\Python\Python39\python.exe "C:/Users/iaak1/Desktop/Leam Python/ASSEX2.py"

Enter the radius of a circle then enter the base and height of a triangle and we will tell your what is left in the circle area if we deduct the area of the triangle. Please be sure you enter values in CM

Please enter the radius

-5

Please enter the radius and be sure you input a positive number

5

Please enter the height

-15

Please enter the height and please be sure you input a positive amount

15

Please enter the base

-25

Please enter the base and ensure you are entering a positive number

25

It seems the area of your circle is -108.95 cm too small for your triangle

Process finished with exit code 0

This is what it does when the triangle area is equal or less than circle.

C:\Users\iaak1\AppData\Local\Programs\Python\Python39\python.exe "C:/Users/iaak1/Desktop/Leam Python/ASSEX2.py"

Enter the radius of a circle then enter the base and height of a triangle and we will tell your what is left in the circle area if we deduct the area of the triangle. Please be sure you enter values in CM

Please enter the radius

-5

Please enter the radius and be sure you input a positive number

5

Please enter the height

-1

Please enter the height and please be sure you input a positive amount

1

Please enter the base

-2

Please enter the base and ensure you are entering a positive number

2

The total area left is 77.55 cm.

Process finished with exit code 0

4.3. Exercise 3

```
# Exercise 3 sees the user input 5 items along with their prices we then
calculate totals including one with
# a 15% staff discount

item_list = [] # here is the item list waiting to be populated
price_list = [] # here is teh price list waiting to be populated
list_size = 5 # the item and price list will not exceed 5 items

while len(price_list) < list_size: # this while loop will run until the
price list has 5 items in it, the price list
    # is used as its the one taking the finaly input out of the two
    item = input("Please enter item name: ")
    item_list.append(item) # here items are added to item list
    price = float(input("now input how much it costs in £ ")) # this is
where items get added to price list
    while price < 0: # this line check that price inputed is not a
negative amount
        price = float(input("Please ensure price is one with a positive
value in £ "))

    price_list.append(price) # this line add the price to the list

print("Products entered \n 1.", item_list[0], "£", price_list[0], "\n 2.",
item_list[1], "£", price_list[1], "\n 3.",
    item_list[2], "£", price_list[2], "\n 4.", item_list[3], "£",
price_list[3], "\n 5.", item_list[4], "£",
    price_list[4],) # here we print a list of said items with prices
next to them calling items along with their
# corresponding prices on spereate lines each item also gets numbered here

staff_dis = 15 # amount that staff discount is out of 100%
total_price = price_list[0] + price_list[1] + price_list[2] + price_list[3]
+price_list[4]
# here total price is calculated
bill_staff_dis = (total_price / 100) * staff_dis # here staff discount is
worked out
bill_after_dis = total_price - bill_staff_dis # here we work out new total
after deducting staff discount

print("The total price is: £", round( total_price, 2 ) ) # here total is
printed
print("The total with staff discount is: £", round( bill_after_dis, 2 ) )
# once staff discount is factored in this is
# amount due being printed
```

C:\Users\iaak1\AppData\Local\Programs\Python\Python39\python.exe "C:/Users/iaak1/Desktop/Leam Python/assex3.py"

Please enter item name: pizza
now input how much it costs in £ 3.45
Please enter item name: chips
now input how much it costs in £ 1.20
Please enter item name: tea
now input how much it costs in £ -1.35
Please ensure price is one with a positive value in £ 1.35
Please enter item name: coffe
now input how much it costs in £ 1.82
Please enter item name: banana
now input how much it costs in £ 0.17
Products entered

Iskander Abd-Al-Kerim - 0250002@student.uwtsd.ac.uk

1. pizza £ 3.45
2. chips £ 1.2
3. tea £ 1.35
4. coffe £ 1.82
5. banana £ 0.17
The total price is: £ 7.99
The total with staff discount is: £ 6.79

Process finished with exit code 0

4.4. Exercise 4

```
# in this version of our noon program we spice things up a little by
# introducing the letters E S and T in order to
# give the user the the option of inputting one of three words for input.
# TOON, NONE or the original NOON! we will
# use a def function for each letter and define spacing for travel between
# different latter types.

import math
import turtle
from turtle import Turtle

hero_hard: Turtle = turtle.Turtle()

def input_number(message): # inputNumber ensures user input a numerical
value and in the event another value is
# entered it prompts for correct input
    while True:
        try:
            userInput = int(input(message))
        except ValueError:
            print("Not an integer! Try again.")
            continue
        else:
            return userInput
            break

# the program starts here by asking the user to input a number between 15
# and 45 to represent the width of the letters,
user_w_width = input_number("Enter a word width between 15 and 45 in order
for us to calculate the "
                            "dimensions of your words: \n")
while user_w_width < 15 or user_w_width > 45: # here a while loop is used
to ensure that the correct number is inputted
    user_w_width = input_number("Try again and this time input a whole
number between 15 and 45: \n")

word_width = user_w_width # n ends is what we base the width of the words
upon word width is based on user w width
word_height = word_width * 5 # the height of the N based on the ends * 5
out_circle = word_height / 2 # outer ring of the O of which the radius is
```

```

based on hlf the N height

move_oo = word_width * 5.5 # jump distance from o to o
move_on = word_width * 3 # jump distance from o to n

def move_turtle():
    hero_hard.penup() # here we reposition our turtle to begging from a
    more suitable place on screen
    hero_hard.back(word_width * 5)
    hero_hard.pendown()

# call N will make turtle draw the letter N
def call_N():
    n_angle1 = 45 # the angle at which we will turn to make the n
    word_height = word_width * 5 # the height of the N based on the ends *
    5
    n_adj = word_width * 3 # inner height of n
    n_opp = word_width * 3 # inner width of n
    result_Nadj_opp = n_adj ** 2 + n_opp ** 2
    n_hyp = math.sqrt(result_Nadj_opp) # refers to length on angled line
    on n
    for my_counter in range(2):
        hero_hard.pendown()
        hero_hard.forward(word_width)
        hero_hard.left(90)
        hero_hard.forward(n_adj)
        hero_hard.right(n_angle1 * 3)
        hero_hard.forward(n_hyp)
        hero_hard.left(n_angle1)
        hero_hard.forward(word_width)
        hero_hard.left(90)
        hero_hard.forward(word_height)
        hero_hard.left(90)

def call_O():
    # o start
    # outer ring
    hero_hard.circle(word_height / 2)
    hero_hard.left(90)
    hero_hard.penup()
    hero_hard.forward(word_width)
    hero_hard.pendown()
    hero_hard.right(90)
    # inner ring
    hero_hard.circle((word_height / 2) - word_width)
    # now we reposition the turtle on the base of the letters
    hero_hard.penup()
    hero_hard.right(90)
    hero_hard.forward(word_width)
    hero_hard.left(90)
    hero_hard.pendown()
    # o end

def call_T():
    # t start
    hero_hard.forward(word_width)
    hero_hard.left(90)

```

```

hero_hard.forward(word_width * 4)
hero_hard.right(90)
hero_hard.forward(word_width * 2)
hero_hard.left(90)
hero_hard.forward(word_width)
hero_hard.left(90)
hero_hard.forward(word_width * 5)
hero_hard.left(90)
hero_hard.forward(word_width)
hero_hard.left(90)
hero_hard.forward(word_width * 2)
hero_hard.right(90)
hero_hard.forward(word_width * 4)
hero_hard.left(90)
# t ends

```

```

def call_E(): # drawing E
    hero_hard.forward(word_width * 4)
    hero_hard.left(90)
    hero_hard.forward(word_width)
    hero_hard.left(90)
    hero_hard.forward(word_width * 3)
    hero_hard.right(90)
    hero_hard.forward(word_width)
    hero_hard.right(90)
    hero_hard.forward(word_width * 3)
    hero_hard.left(90)
    hero_hard.forward(word_width)
    hero_hard.left(90)
    hero_hard.forward(word_width * 3)
    hero_hard.right(90)
    hero_hard.forward(word_width)
    hero_hard.right(90)
    hero_hard.forward(word_width * 3)
    hero_hard.left(90)
    hero_hard.forward(word_width)
    hero_hard.left(90)
    hero_hard.forward(word_width * 4)
    hero_hard.left(90)
    hero_hard.forward(word_width * 5)
    hero_hard.left(90)

```

```

def move_NO(): # moving from N to O
    hero_hard.penup()
    hero_hard.forward(word_width * 8)
    hero_hard.pendown()

```

```

def move_OO(): # moving from O to O
    hero_hard.penup()
    hero_hard.forward(word_width * 5.5)
    hero_hard.pendown()

```

```

def move_ON(): # moving from O to N
    hero_hard.penup()
    hero_hard.forward(word_width * 3)
    hero_hard.pendown()

```

```

def move_TO(): # moving from T to O
    hero_hard.penup()
    hero_hard.forward(word_width * 6)
    hero_hard.pendown()

def move_NE(): # moving from N to E
    hero_hard.penup()
    hero_hard.forward(word_width * 6)
    hero_hard.pendown()

def pick_word(): # Here we create a function called pick_word that allows
    # if statements we look at calling the letter drawing function with the
    # are printed. so user if word is not listed user is offered option to
    # try inputting a valid word to be drawn again
    # otherwise program will end
    print("Input either NOON, TOON or NONE in capital letters and i will
draw it out for you!")
    user_word = input()
    if user_word == 'NOON':
        print(move_turtle(), call_N(), move_NO(), call_O(), move_OO(),
call_O(), move_ON(), call_N())
    elif user_word == 'TOON':
        print(move_turtle(), call_T(), move_TO(), call_O(), move_OO(),
call_O(), move_ON(), call_N())
    elif user_word == 'NONE':
        print(move_turtle(), call_N(), move_NO(), call_O(), move_ON(),
call_N(), move_NE(), call_E())
    else:
        print("Invalid input!")

    again()

# here we define the try inputting word again function

def again():
    call_again = input('Do you want to try again?
Please type Y for YES or N for NO.
')
    if call_again.upper() == 'Y':
        # the program starts here by asking the user to input a number
        # between 15 and 45 to represent the
        # width of the letters,
        user_w_width = input_number("Enter a word width between 15 and 45
in order for us to calculate the "
dimensions of your words: \n")
        while user_w_width < 15 or user_w_width > 45: # here a while loop
is used to ensure that the correct
        # number is inputted
        user_w_width = input_number("Try again and this time input a
whole number between 15 and 45: \n")
        print(hero_hard.penup(), hero_hard.back(word_width * 15),
hero_hard.pendown(), hero_hard.clear(),
pick_word()) # if user chooses option to input more words
        # firstly turtle is moved back close to its original starting point

```

```

and then everything the turtle has
# previously done on screen is cleared
elif call_again.upper() == 'N':
    print('See you later.')
else:
    again()

pick_word()

turtle.done()

```

C:\Users\iaak1\AppData\Local\Programs\Python\Python39\python.exe "C:/Users/iaak1/Desktop/Learn Python/assexx.py"

Enter a word width between 15 and 45 in order for us to calculate the dimensions of your words:

25

Input either NOON, TOON or NONE in capital letters and i will draw it out for you!

non

Invalid input!

Do you want to try again?

Please type Y for YES or N for NO.

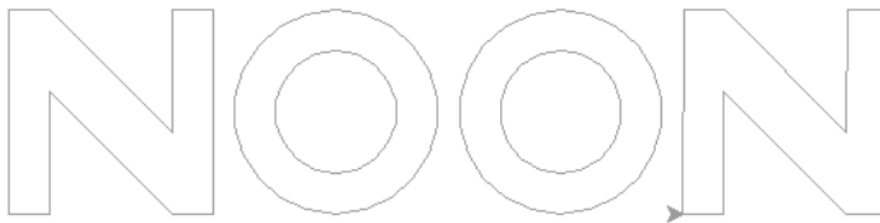
Y

Enter a word width between 15 and 45 in order for us to calculate the dimensions of your words:

25

Input either NOON, TOON or NONE in capital letters and i will draw it out for you!

NOON



None None None None None None None None

Do you want to try again?

Please type Y for YES or N for NO.

Y

Enter a word width between 15 and 45 in order for us to calculate the dimensions of your words:

15

Input either NOON, TOON or NONE in capital letters and i will draw it out for you!

TOON

TOON

None None None None None None None None

Do you want to try again?

Please type Y for YES or N for NO.

Y

Enter a word width between 15 and 45 in order for us to calculate the dimensions of your words:

NONE

Not an integer! Try again.

Enter a word width between 15 and 45 in order for us to calculate the dimensions of your words:

35

Input either NOON, TOON or NONE in capital letters and i will draw it out for you!

NONE

NONE

None None None None None None None None

Do you want to try again?

Please type Y for YES or N for NO.

N

Do you want to try again?

Please type Y for YES or N for NO.

See you later.

None None None None None

None None None None None

None None None None None

5. Conclusion

This was a brief introduction to computer programming the code really is ultimately the strongest tool in the process. However, many tools were used to make this report possible besides the language python from <https://www.python.org/> to PyCharm from <https://www.jetbrains.com/pycharm/> figuring out digital solutions is possible. There is a process involved but the saying where there's a will there is a way is certainly applicable to coding.

References

- [guyhaas, “What Is Computer Programming?,” [Online]. Available:
1 <http://guyhaas.com/bfoit/itp/Programming.html#:~:text=Computer%20programming%20involves%20develo>
] <ping%20lists,%2C%20sounds%2C%20etc....> [Accessed 15 04 2021].
[icarus.cs.weber.edu, “4.2 Variables And Memory Addresses,” [Online]. Available:
2 https://icarus.cs.weber.edu/~dab/cs1410/textbook/4.Pointers/vars_address.html#:~:text=Any%20programmi
] [ng%20element%20\(variable%2C%20object,and%20stored%20in%20a%20pointer..](ng%20element%20(variable%2C%20object,and%20stored%20in%20a%20pointer..) [Accessed 19 03 2021].
[C. SINGH, “Pointers in C Programming with examples,” C-PROGRAMMING, [Online]. Available:
3 [https://beginnersbook.com/2014/01/c-pointers/.](https://beginnersbook.com/2014/01/c-pointers/) [Accessed 24 04 2021].
]
[w3schools, “Python Variables,” [Online]. Available:
4 https://www.w3schools.com/python/python_variables.asp#w3sc. [Accessed 24 04 2021].
]
[A. Jama, “Variables and Containers — Python 101,” 20 12 2017. [Online]. Available:
5 [https://medium.com/apjama/variables-and-containers-python-101-](https://medium.com/apjama/variables-and-containers-python-101-1bca7132b124#id_token=eyJhbGciOiJSUzI1NiIsImtpZCI6IjZhOGJhNTY1MmE3MDQ0MTIxZDRmZWRhYzhmMTRkMTRjNTRlNDg5NWllLCJ0eXAiOiJKV1QiQfQyJpc3MiOiJodHRwczovL2FjY291bnRzLmdvb2dsZS5jb20iLCJuYmYiOiJlcyJxNzI4)
] [1bca7132b124#id_token=eyJhbGciOiJSUzI1NiIsImtpZCI6IjZhOGJhNTY1MmE3MDQ0MTIxZDRmZWRhYzhmMTRkMTRjNTRlNDg5NWllLCJ0eXAiOiJKV1QiQfQyJpc3MiOiJodHRwczovL2FjY291bnRzLmdvb2dsZS5jb20iLCJuYmYiOiJlcyJxNzI4](1bca7132b124#id_token=eyJhbGciOiJSUzI1NiIsImtpZCI6IjZhOGJhNTY1MmE3MDQ0MTIxZDRmZWRhYzhmMTRkMTRjNTRlNDg5NWllLCJ0eXAiOiJKV1QiQfQyJpc3MiOiJodHRwczovL2FjY291bnRzLmdvb2dsZS5jb20iLCJuYmYiOiJlcyJxNzI4). [Accessed 19 03 2021].
[w3schools, “trypython.asp?filename=demo_variables_cs,” w3schools, [Online]. Available:
6 https://www.w3schools.com/python/trypython.asp?filename=demo_variables_cs. [Accessed 23 04 2021].
]
[V. Beal, “Variable,” [Online]. Available: [https://www.webopedia.com/definitions/variable/.](https://www.webopedia.com/definitions/variable/) [Accessed 01 04
7 2021].
]
[w3schools.com, “Python If ... Else,” [Online]. Available:
8 https://www.w3schools.com/python/python_conditions.asp. [Accessed 02 04 2021].
]
[o. P. Adewumi*, “Inner loop program construct: A faster way forprogram execution,” 01 08 2018. [Online].
9 Available: <https://www.degruyter.com/document/doi/10.1515/comp-2018-0004/html>. [Accessed 02 04
] 2021].
[rosettaocode, “Loops/For with a specified step,” 31 03 2021. [Online]. Available:
1 http://www.rosettaocode.org/wiki/Loops/For_with_a_specified_step#Python. [Accessed 02 04 2021].
0
]
[w3schools, “Python For Loops,” [Online]. Available:
1 https://www.w3schools.com/python/python_for_loops.asp. [Accessed 02 04 2021].
1
]

FINAL GRADE

75/100

GENERAL COMMENTS

Instructor**PART A**

1. Variables - Good, you just need to make sure that you are clear that Python infers the type unless it is cast. Also, the weak-typing means that it is easier to change a variable from one type to another 4/5

2. If/Else - The word you are looking for here is "decision". Good explanation otherwise with suitable code examples 4.5 /5

3. Loops - You need to distinguish between the two types of loop (for and while) and why they are used in different circumstances. You do allude to it in your examples though! 3.5/5

Part A - 12/15

PART B**Exercise 1**

Logical Stages/Testing - I can't see any logical stages, only testing. The testing is there, but it is not presented too well. You need to state what you are testing, and whether the program behaved as expected. 2 /5

Coding/Annotation - You don't really need all the elifs 1-5 as they are all the same. You could just say if `door_count >= 1` or `door_count <= 5` Other than that, the validation and annotations are very good! 9/10

Exercise 2

Logical Stages/Testing - No logical stages. Testing is better here, but don't forget to say whether the result was as expected or not. 5/10

Coding/Annotation - Good code and validation. You might want to think about relative sizes of the triangle and the radius of the circle you have also considered that the triangle cannot be too big for the circle. Very good. 10/10

Exercise 3

Logical Stages/Testing - No logical stages shown. Again, testing needs better labelling. 4.5/10

Coding/Annotation - Code is good with validation of the inputs to the list (ie not negative) printing of the list and adding of the list could have been looped to allow for easier expansion/efficiency 13/15

Exercise 4 (Turtle)

Logical Stages/Testing - No logical stages. Testing OK, don't forget to label outcome 5/10

Coding/Annotation - A very good attempt, you do have a loop, and also even different letters. Functions used. You could have use a loop on the N as it has rotational symmetry, but this is a minor point. Very good effort! 14/15
A very good assignment, just be careful to put in everything asked for in the assignment brief, ie the logical stages.

TOTAL: 74.5%

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19
