

Software Development - Alex Tzafos

by Alexander Tzafos

Submission date: 29-Apr-2022 12:40PM (UTC+0100)

Submission ID: 177811742

File name: 33391_Alexander_Tzafos_Software_Development_-_Alex_Tzafos_410175_69758880.docx
(514.89K)

Word count: 1930

Character count: 8533

Software Development

Lecturer: Gaynor Thomas
Course: Software Development
Module Code: ACCB4017

Table of Contents

Part A 2

 A1..... 2

 A2..... 2

 A3..... 3

Part B 5

 Exercise 1..... 5

 Exercise 2..... 6

 Exercise 3..... 9

Part A

A1

Variables are used widely throughout the world of computer programming and across all the different platforms. They are used to store information which can be referenced and manipulated if needs be. Alongside this you the user is able to call the variables in your own way so it will make it easier for the program to call upon it later on in your code, however there are a few rules you will need to follow when naming said variables. For example, [1] the variable name can be any length and can consist of uppercase and lowercase letters, digits (0-9) and underscore character. In addition to this the variable name cannot start with a digit. If you then wanted to use more than one word as your variable there are three different layouts you can choose between.

- Camel Case – This is when the first letter in each word is capitalised apart from the first word. (exampleOfCamelCase)
- Pascal Case – This is almost identical to camel case; however, the first word is capitalised. (ExampleOfPascalCase)
- Snake Case – This is when each word is separated by using underscores. (example_of_snake_case)

Furthermore, there are more version of variables in Python and there are static and dynamic. A static attribute will be defined once for the class and is then shared by all instances, for example if used **total = 2** then this would be static and the total has been set. Whereas a dynamic attribute is defined for individual instances only, for example if you had **total = x** then this would be dynamic as the x value can change.

A2

An if statement in Python is used if you want to evaluate if a statement is true or false. Once you have entered in your required statement it will begin going through each step of your code and if the if statement is executed then it is true, however it is found to be false the else statement will be executed.

```
print("What would you like for breakfast? ")
choice = input()

if choice == "eggs":
    print("scrambled or fried")
    subChoice = input().lower() # makes it lower
    if subChoice == "scrambled":
        print("Scrambled eggs are coming")
    elif subChoice == "fried":
        print("Fried it is")
    else:
        print("We don't have anything else")
```

This screenshot that you can see above is of an if/else statement that I had made. When you run the program, you will be asked "What would you like for breakfast?", if you typed in eggs, you would then enter the if statement and asked, "scrambled or fried". If the user was to enter either of them two options then the statement will be true and the if statement is run, however if the user enters anything else the statement would then become false and the else statement will be run.

A3

In python there are two main types of loops and they are for and while. I have used both, and I will provide examples for them both. The main difference between while and for is that with a for loop the number of iterations already know and is used to find out a certain result, whereas with a while loop the command will constantly run until either the statement is false or a condition is met. In the world of computer programming a loop is a sequence of different instructions that is constantly repeated until a condition is met. Usually, a process is done such as getting an item like a piece of data and changing it repeatedly. Once all the conditions have been reached the next step is to exit the code which is done automatically. Loops are needed in programming for many different things, for example you could make a counter where it would count to whatever number you choose, you could make a guessing game where the user would have to guess a random number and the program would constantly loop until they guess the number correctly.

```
print("Let's count up, we'll start at 1 and go up to 10")

myCounter = 1

while myCounter <= 10:
    print("Counting...", myCounter)
    myCounter += 1
```

This screenshot that you can see is of a while loop I had done. The purpose of this was to count from 1 to 10 using a loop. Once the code is run the message would display on the screen saying 'Counting... 1', 'Counting...2' etc, all the way up to 10. After the code had reached 10 it would exit as the statement would then be false.

```
for counter in range(1, 16, 1):  
    total = counter * 7  
    print(counter, "times 7 is", total)  
  
print("End of program")
```

This screenshot is an example of a simple for loop that showcases the 7 times table up to 16. Within this for loop I have a set a range that goes from 1 – 16 and will increase in increments of 1 until it has gone through 16 times. The program will start with '1 times 7 is 7' and will work it way up to 16 times 7. Once the code has gone through all 16 possibilities it will then end the code.

Part B

Exercise 1

Logical stage

- 1) Prompt the user to enter name
- 2) Check if the user has entered in a valid name (if not back to 1)
- 3) Prompt user to enter age
- 4) Check if the user has entered in a valid age (if not back to 3)
- 5) Prompt user to enter height
- 6) Check if the user has entered in a valid height (if not back to 5)
- 7) Print results

Program

```
# program for a user to enter their details
print("Hello welcome to my program, please enter your name: ")
EnterName = str(input()) # user enters their name

print("Please enter your age: ")
EnterAge = int(input()) # user enters their age

print("Please enter your height in metres: ")
EnterHeight = float(input()) # user enters their height

print("Thanks, you entered: ", EnterName, "-", EnterAge, "years old", "-",
      EnterHeight, "cm tall")
# print out the results
```

Testing

```
Hello welcome to my program, please enter your name:
Alex
Please enter your age:
21
Please enter your height in metres:
1.98
Thanks, you entered: Alex - 21 years old - 1.98 cm tall
```

The program got the expected results

Exercise 2

Logical

- 1) Users enters in radius of circle
- 2) Checks to see if its bigger than zero (if not back to 1)
- 3) Users enters in one side of the square
- 4) Checks to see if its smaller than the radius (if not back to 3)
- 5) Users enters in width of the rectangle
- 6) Checks to see if its smaller than radius (if not back to 5)
- 7) Users enters in length of the rectangle
- 8) Check to see if its smaller than radius (if not back to 7)
- 9) Works out area of the circle using $A = \pi r^2$
- 10) Works out area of square and rectangle using $A = l * w$
- 11) Takes away the area of the circle from the sum of the area of the square and rectangle to work out the area of the shaded circle
- 12) Prints out result

Code

```
# Program to find ou the area of shaded circle when from the users
dimensions
print("Lets work out the area of some shapes")

print("Enter radius of the circle: ")
radius = float(input())      # user enters radius of circle

while (radius <=0) : # checks to see if radius is bigger than zero
    print("Try again, must be more than 0")
    radius = float(input())  # try again

print("Enter side of the square: ")      # user enters side of square
side = float(input())

while (side > radius) : # checks to see if the side is bigger than the
radius
    print("Try again, must be smaller than the radius")
    side = float(input())  # try again

print("Enter width of the rectangle: ")  # user enters width of the
rectangle
width = float(input())

while (width > radius) : # checks to see if the width is bigger than the
radius
    print("Try again, must be smaller than the radius")
    width = float(input())  # try again

print("Enter length of the rectangle: ")  # user enters length of
rectangle
length = float(input())
```

```
while (length > radius) : # checks to see if the length is bigger than the
radius
    print("Try again, must be smaller than the radius")
    radius = float(input()) # try again

AreaOfCircle = math.pi*(radius **2) # works out the area of the circle
AreaOfSqaure = side * side # works out the area of the square
AreaOfRectangle = length * width # works out the area of the rectangle
AreaOfShadedArea = AreaOfCircle - (AreaOfRectangle + AreaOfSqaure) # works
out the ares of the shaded area

print("The shaded area is", AreaOfShadedArea,"cm") # prints results
```

Testing

```
Lets work out the area of some shapes
Enter radius of the circle:
10
Enter side of the square:
2
Enter width of the rectangle:
3
Enter length of the rectangle:
4
The shaded area is 298.1592653589793 cm
```

Correct results, program has worked

```
Enter radius of the circle:
-1
Try again, must be more than 0
```

Testing the boundaries, code works perfectly

```
Lets work out the area of some shapes
Enter radius of the circle:
10
Enter side of the square:
12
Try again, must be smaller than the radius
|
```

Testing the boundaries, code works perfectly

```
Lets work out the area of some shapes
Enter radius of the circle:
12
Enter side of the square:
4
Enter width of the rectangle:
12
Enter length of the rectangle:
13
Try again, must be smaller than the radius
|
```

Testing the boundaries, code works perfectly

Exercise 3

Logical

- 1) Users enters miles for Monday
- 2) Checks to see if its bigger than zero (if not back to 1)
- 3) Users enters miles for Tuesday
- 4) Checks to see if its bigger than zero (if not back to 3)
- 5) Users enters in miles for Wednesday
- 6) Checks to see if its bigger than zero (if not back to 5)
- 7) Users enters in miles for Thursday
- 8) Checks to see if its bigger than zero (if not back to 7)
- 9) User enters in miles for Friday
- 10) Checks to see if its bigger than zero (if not back to 9)
- 11) User enters in miles for Saturday
- 12) Checks to see if its bigger than zero (if not back to 11)
- 13) User enters in miles for Sunday
- 14) Checks to see if its bigger than zero (if not back to 13)
- 15) Works out sum of miles for the week
- 16) Prints result
- 17) Works out total cost of fuel for the week
- 18) Prints results in £

Code

```
# A program to work out fuel costs
print("Enter miles for Monday: ")
monday = float(input()) # user enters miles for monday
while (monday <= 0) : # checks to see if amount is bigger than zero
    print("Must be greater than zero")
    monday = float(input()) # try again

print("Enter miles for Tuesday: ")
tuesday = float(input()) # user enters miles for tuesday
while (tuesday <= 0) : # checks to see if amount is bigger than zero
    print("Must be greater than zero")
    tuesday = float(input()) # try again

print("Enter miles for Wednesday: ")
wednesday = float(input()) # user enters miles for wednesday
while (wednesday <= 0) : # checks to see if amount is bigger than zero
    print("Must be greater than zero")
    wednesday = float(input()) # try again

print("Enter miles for Thursday: ")
thursday = float(input()) # user enters miles for thursday
while (thursday <= 0) : # checks to see if amount is bigger than zero
    print("Must be greater than zero")
    thursday = float(input()) # try again
```

```
print("Enter miles for Friday: ")
friday = float(input()) # user enters miles for friday
while (friday <= 0) : # checks to see if amount is bigger than zero
    print("Must be greater than zero")
    friday = float(input()) # try again

print("Enter miles for Saturday: ")
saturday = float(input()) # user enters miles for saturday
while (saturday <= 0) : # checks to see if amount is bigger than zero
    print("Must be greater than zero")
    saturday = float(input()) # try again

print("Enter miles for Sunday: ")
sunday = float(input()) # user enters miles for sunday
while (sunday <= 0) : # checks to see if amount is bigger than zero
    print("Must be greater than zero")
    sunday = float(input()) # try again

TotalMileage = monday + tuesday + wednesday + thursday + friday + saturday
+ sunday # works out the sum of miles
print("The total mileage was", TotalMileage) # prints result

CostOfPetrol = TotalMileage / 11.2 * 1.35 # works out cost of fuel for
the week
print("The cost of petrol: £", CostOfPetrol) # prints results
```

Testing

```
Enter miles for Monday:
0
Must be greater than zero
2
Enter miles for Tuesday:
0
Must be greater than zero
2
Enter miles for Wednesday:
0
Must be greater than zero
2
Enter miles for Thursday:
0
Must be greater than zero
2
Enter miles for Friday:
0
Must be greater than zero
2
Enter miles for Saturday:
0
Must be greater than zero
2
Enter miles for Sunday:
0
Must be greater than zero
2
```

Testing the boundaries by entering zero, code works perfect

```
Enter miles for Monday:
10
Enter miles for Tuesday:
10
Enter miles for Wednesday:
10
Enter miles for Thursday:
4
Enter miles for Friday:
6
Enter miles for Saturday:
7
Enter miles for Sunday:
9
The total mileage was 56.0
The cost of petrol: £ 6.75
```

Perfect piece of code

Software Development - Alex Tzafos

GRADEMARK REPORT

FINAL GRADE

41 / 100

GENERAL COMMENTS

Instructor

PART A

Variables - You have talked about variables, but you haven't talked about types - you need to say what the types are ie int, float string. You also need to talk about the specific way that Python infers the variable type and how/why these types are considered dynamic (in Python, types can change within the lifetime of the program, for example, when dividing two integers, the resulting type is changed to float) 1.5/5

If/Else - You've got the main point but you also need to talk about multiple conditions eg if/elif/else and conditions that have two pieces of logic (using and/or) 2.5/5

Loops - You've got the main points with suitable examples. 5/5

PART B

Exercise 1

Logical Stages-Testing - Logical stages good. Testing only tests the happy path (see my comment about validation below) 3.5/5

Coding-Annotation - A working program, but you need validation here to ensure that the user does not enter

out of range age/height etc. (ie reasonable range of human life span/height) 5/10

Exercise 2

Logical Stages-Testing - Good logical stages. Testing OK, could be a little wider 4 /5

Coding-Annotation - Good program, you consider size of circle against the square/rectangle, could also consider size of square/rectangle against each other. Also need to validate for minus numbers on squares and rectangle 8/10

Exercise 3

Logical Stages-Testing - Logical stages are OK. Testing is OK, but limited by coding (see below 6/10

Coding-Annotation -Working program produced, but lists and loops used - using a list was the whole point of this exercise (it's in the question!) 5/10

Exercise 4

Not attempted

General comment: All assignments should really have a short Introduction and a conclusion - just to set the scene on what the assignment is about and in the conclusion, to discuss how things went and what you would do differently next time. There are no marks assigned for that in this particular assignment, but it is good academic practice.

Assignment overall: 40.5%

Test mark: 47.5%

OVERALL MARK FOR MODULE: 43%

PAGE 1
PAGE 2
PAGE 3
PAGE 4
PAGE 5
PAGE 6
PAGE 7
PAGE 8
PAGE 9
PAGE 10
PAGE 11
PAGE 12