

Programming

by James Philip Chappel

FILE	26591_JAMES_PHILIP_CHAPPEL_PROGRAMMING_341067_970126913.DOCX (239.48K)		
TIME SUBMITTED	27-APR-2021 02:53PM (UTC+0100)	WORD COUNT	1342
SUBMISSION ID	150706438	CHARACTER COUNT	6960

Introduction to computer programming

James Chappel 2015997

April 27th 2021

Part A

A1. A variable is a named unit of data that is assigned a value. They are important in writing programs because they are easy to change when required. They are the language that communicates with the PC. 399

Python is strongly, dynamically typed:

- **Strong** typing means that the type of a value doesn't change in unexpected ways. A string containing only digits doesn't magically become a number, as may happen in Perl. Every change of type requires an explicit conversion.
- **Dynamic** typing means that runtime objects (values) have a type, as opposed to static typing where variables have a type.

A2. An if/else statement is used when a decision needs to be made within a program. It allows the user to choose a pathway depending on their input.

Here is an example of how an if statement can be used in python:

1. Write a program to allow the user to enter a number between 1 and 20. The program should tell the user whether the number they entered was 10 or more, or less than 10.

Sample output:

```
Please enter your number:
12
In the top half (10 or more)
```

```
Please enter your number:
7
Below 10
```

Solution:

```
print("Please enter your number: ")
userNumber = int(input())

if userNumber >= 10:
    print("In the top half (10 or more)")
else:
    print("Below 10")
```

A3. Loops are required when a section of code needs to be repeated. A while loop could be very usual when a count up or countdown is required. A for loop might be used when creating a times table for example, each number can be automatically calculated instead of going through each number individually.

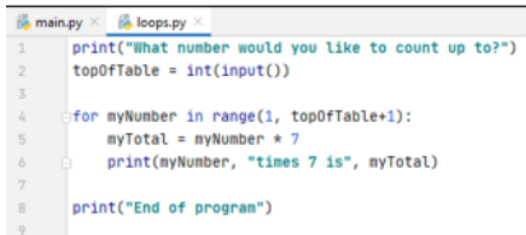
Here is an example of a while loop in python to verify that someone is 18 and above:

```
print("Enter your age, it must be 18 or more: ")
userAge = int(input())

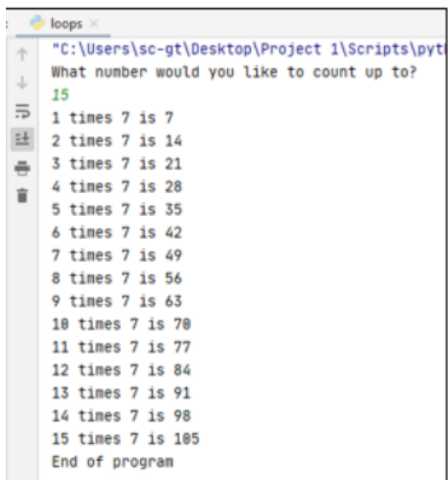
while userAge < 18:
    print("Sorry that was invalid")
    userAge = int(input("Try again: "))

print("Thank you!")
```

Here is an example of a for loop being used in python to calculate a times table:

A screenshot of a Python IDE with two tabs: 'main.py' and 'loops.py'. The 'loops.py' tab is active, showing a script that asks for a number to count up to and then prints a multiplication table for 7. The script is as follows:

```
1 print("What number would you like to count up to?")
2 topOfTable = int(input())
3
4 for myNumber in range(1, topOfTable+1):
5     myTotal = myNumber * 7
6     print(myNumber, "times 7 is", myTotal)
7
8 print("End of program")
9
```

A screenshot of a Python IDE showing the output of the script from the previous block. The output is displayed in a console window with the following text:

```
"C:\Users\sc-gt\Desktop\Project 1\Scripts\pyt
What number would you like to count up to?
15
1 times 7 is 7
2 times 7 is 14
3 times 7 is 21
4 times 7 is 28
5 times 7 is 35
6 times 7 is 42
7 times 7 is 49
8 times 7 is 56
9 times 7 is 63
10 times 7 is 70
11 times 7 is 77
12 times 7 is 84
13 times 7 is 91
14 times 7 is 98
15 times 7 is 105
End of program
```

1.

Logical stages:

1. Prompt user to enter car make.
2. Prompt user to enter car model.
3. Prompt user to enter car milage.
4. Cast to float as user could input to 2 decimal places.
5. Prompt user to enter number of car doors.
6. Enter the type to determine the class of the variable.

2. Code with annotation for exercise 1.

```
# Program to input the details of car
print("enter the car make:")                # prompt
user to enter car make
carMake = input()
print("enter the car model:")              # prompt
user to enter car model
carModel = input()
print("enter the car mileage:")            # prompt
user to enter mileage
carMile = float(input())                   # Cast as
float

while carMile <=0:
    print("try again")
    carMile = float(input())

print("enter the number of doors:")
doorNo = int(input())

print(type(carMake))                      #
Determine class of variables
print(type(carModel))
print(type(carMile))
print(type(doorNo))
```

Testing

normal within range

```
C:\Users\jchap\PycharmProject>
enter the car make:
Ford
enter the car model:
Focus
enter the car milage:
55000
enter the number of doors:
4
```

Boundaries

Test milage for valid input.

```
enter the car make:
Ford
enter the car model:
Focus
enter the car milage:
0
try again
|
```

Test out of range values

```
Ford
enter the car model:
Focus
enter the car milage:
-1
try again
100
enter the number of doors:
|
```

2.

Logical stages

1. Prompt user to enter the base of the triangle.
2. Use while loop to validate user enter minimum value.
3. Prompt use to enter the height of the triangle.
4. Input while loop to validate the user input.
5. Input formula to calculate the area of the triangle.
6. Prompt user for the radius.
7. Input while loop to validate the user to input a minimum value.
8. Input variable for pi.
9. input the formula for the area of a circle.
10. round the number to 2 decimal places.
11. Input formula for the area of the shaded area.
12. Round the total sum to 2 decimal places.

2.Code with annotation for exercise 2.

```
print("Enter the base: ") # prompt user to
enter base
base = float(input())

while base <= 0 or base > 10: # while loop
    validates user input minimum values
    print("sorry try again")
    base = float(input())

print("enter the height:") # prompt user for
the height
height = float(input())

while height <= 0 or height > 10: # while loop
    validates user input minimum values
    print("sorry try again")
    height = float(input())

area = (base * height) / 2 # Input formula
to calculate area of triangle

print("The area of the triangle is: ", area, "cm")

print("enter the radius") # prompt user for
radius
radius = float(input())

while radius <= 0 or radius > 8: # while loop
    validates user input minimum values
```

```

    print("sorry try again")
    radius = float(input())

pi = 3.14 #
create variable for pi
circ_area = pi * radius * radius #
formula for area of circle
print("the area of the circle is:", round(circ_area, 2), "cm") #
round number to 2 decimal places

tot_sum = circ_area - area #
formula for area of shaded area
print("the total of shaded area is:", round(tot_sum, 2), "cm") #
round number to 2 decimal places

```

Testing

Normal path

```

C:\Users\jchap\PycharmProjects\pythonProject\venv\Scripts\python.exe
Enter the base:
5
enter the height:
6
The area of the triangle is: 15.0 cm
enter the radius
|

```

```

6
The area of the triangle is: 15.0 cm
enter the radius
8
the area of the circle is: 200.96 cm
the total of shaded area is: 185.96 cm

Process finished with exit code 0
|

```

Boundaries

Test base for valid input

```
C:\Users\jchap\PycharmProjects\pythonProject\venv>
Enter the base:
0
sorry try again
0
sorry try again
1
enter the height:
|
```

Test height for valid input

```
enter the height:
0
sorry try again
0
sorry try again
5
The area of the triangle is:  2.5 cm
enter the radius
|
```

Test for out of range numbers

```
C:\Users\jchap\PycharmProjects\pythonProject>
Enter the base:
-2
sorry try again
3
enter the height:
-2
sorry try again
|
```

3.

Logical stages

1. Create variable for veg and price.
2. Prompt user for item 1.
3. Add item to the list.
4. Prompt user for item 1 price.
5. Add the item price to list.
6. Prompt user for item 2
7. Add item to the list.
8. Prompt user for item 2 price.
9. add the item price to the list.
10. Prompt the user for item 3.
11. add the item to the list.
12. Prompt the user for item 3 price.
13. Add the item price to the list.
14. Prompt the user for item 4.
15. Add the item to the list.
16. Prompt the user for item 4 price.
17. Add the item price to the list.
18. Prompt the user for item 5.
19. Add the item to the list.
20. Prompt the user for item 5 price.
21. Add the price to the list.
22. Display the two lists side by side.
23. input formula to calculate the total of the list.
24. Input formula to calculate the 15% discount.
25. Display total sum and 15% discount.

Code with annotation for exercise 3.

```
veg=[] # Create variable for veg and price
price=[]
print("Enter item: 1") # Prompt user for item 1
veg.append(input()) # Adds item to list
print("Enter item 1 price") # Prompt user for item 1 price
price.append(float(input())) # Adds to list
print("Enter item 2:") # Prompt user for item 2
veg.append(input()) # Adds item to list
print("Enter item 2 price:") # Prompt user for item 2 price
price.append(float(input())) # Adds item to list
print("Enter item 3:") # Prompt user for item 3
veg.append(input()) # Adds to list
print("Enter item 3 price:") # Prompt user for item 3 price
price.append(float(input())) # Adds to list
print("Enter item 4:") # Prompt user for item 4
veg.append(input()) # Adds to list
print("Enter item 4 price:") # Prompt user for item 4 price
price.append(float(input())) # Adds to list
print("Enter item 5:") # Prompt user for item 5
veg.append(input()) # Adds to list
print("Enter item 5 price:") # Prompt user for item 5 price
price.append(float(input())) # Adds to list

print("products entered:")

print(veg[0], price[0]) # Displays list of items and prices
print(veg[1], price[1])
print(veg[2], price[2])
print(veg[3], price[3])
print(veg[4], price[4])

total price = price[0] + price[1] + price[2] + price[3] + price[4] # sum
for total of list
staff discount = total price/100 * 85 # sum
for 15% discount
print("Total price is:£", total price)
print("Total with staff discount is:£", round(staff discount, 2))
```

Testing.

Normal pathway

```
C:\Users\jchap\PycharmProjects\py
Enter item: 1
Bananna
Enter item 1 price:
1.00
Enter item 2:
Grapes
Enter item 2 price:
|
```

```
Grapes
Enter item 2 price:
1.30
Enter item 3:
Cheese
Enter item 3 price:
2.20
Enter item 4:
Yogurt|
```

```
2.20
Enter item 4:
Yogurt
Enter item 4 price:
1.35
enter item 5:
Cereal
Enter item 5 price:
3.50
```

```
products entered:
Banannas 1.0
Grapes 1.3
Cheese 2.2
Yogurt 1.35
Cereal 3.5
Total price is:£ 9.35
Total with staff discount is:£ 7.95
```

4.

Logical stages.

1. Import Turtle library
2. Set the speed of the turtle.
3. Lift penup to move turtle.
4. Put pendown to start the letter N.
5. Once N drawn, penup to move turtle.
6. Use for loop to create duplicate O's.
7. Penup to move turtle to the last letter.
8. Pendown to complete letter N.

2. Code with annotation.

```
import turtle                                # Imports Turtle library

myTur = turtle.Turtle()
myTur.speed(10)                             # sets speed of turtle
x=0

myTur.left(180)
myTur.penup()                               # takes the pen away
myTur.forward(270)
myTur.pendown()                             # bring the pen back
myTur.forward(20)                           # start the drawing of the N
myTur.right(90)
myTur.forward(100)
myTur.right(90)
myTur.forward(20)
myTur.right(45)
myTur.forward(100)
myTur.left(135)
myTur.forward(70)
myTur.right(90)
myTur.forward(20)
myTur.right(90)
myTur.forward(100)
myTur.right(90)
myTur.forward(25)
myTur.right(45)
myTur.forward(90)
myTur.left(135)
myTur.forward(64)
myTur.left(90)
myTur.penup()                               # end of N ?
myTur.forward(170)
myTur.pendown()                             # start of O loop

for x in range(0, 2):                       # for x in range(0, 2) loops circle
twice
    myTur.circle(55)
```

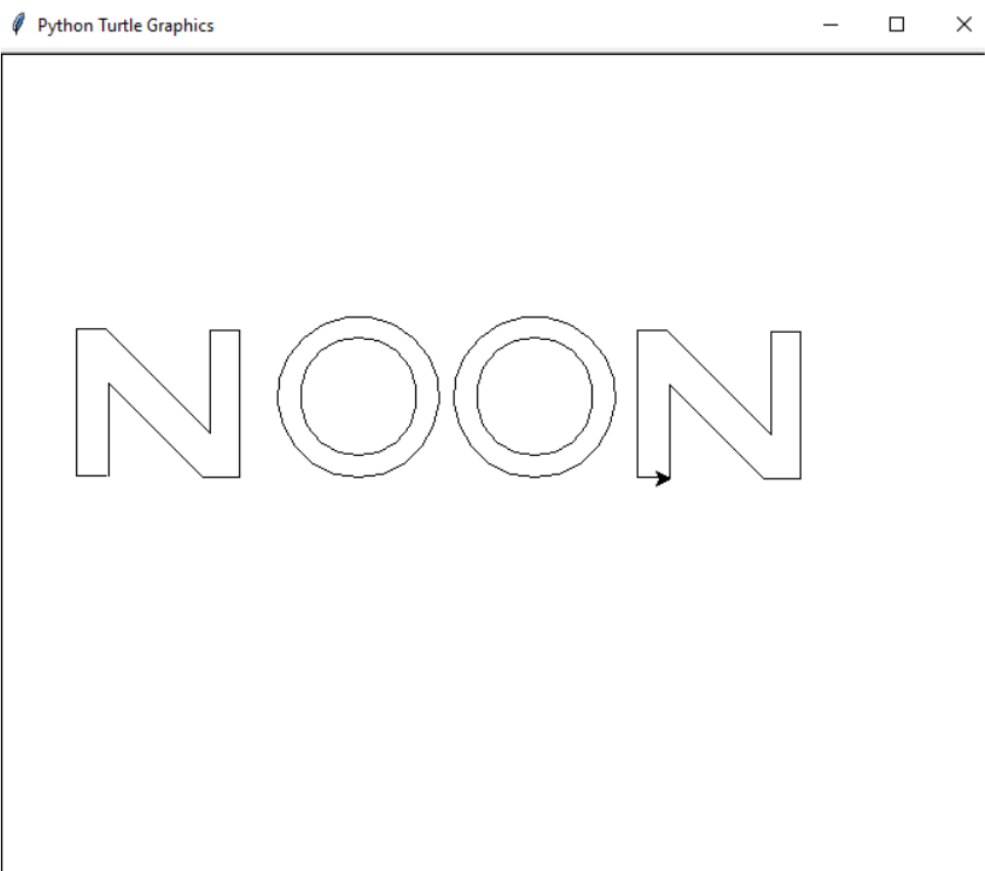
```
myTur.left(90)
myTur.penup()
myTur.forward(15)
myTur.pendown()
myTur.right(90)
myTur.circle(40)
myTur.right(90)
myTur.penup()
myTur.forward(15)
myTur.left(90)
myTur.forward(120)
myTur.pendown()

myTur.penup()
myTur.right(180)
myTur.forward(30)

myTur.pendown()
myTur.forward(20)           # Completes the last N
myTur.right(90)
myTur.forward(100)
myTur.right(90)
myTur.forward(20)
myTur.right(45)
myTur.forward(100)
myTur.left(135)
myTur.forward(70)
myTur.right(90)
myTur.forward(20)
myTur.right(90)
myTur.forward(100)
myTur.right(90)
myTur.forward(25)
myTur.right(45)
myTur.forward(90)
myTur.left(135)
myTur.forward(64)          # end of N
myTur.left(90)
myTur.penup()

turtle.done()              # End of program
```

Output result.



FINAL GRADE

72/100

GENERAL COMMENTS

Instructor

PART A

1. Variables - You are not quite right and need to look at sources a bit wider than Stack overflow. Python is dynamically typed, but it does not always require explicit conversion and types can change (for example, as we have seen when dividing two integers) You need more expansion of the dynamic typing area you mention. Also, give some examples of the types available eg str, int, float 2.5/5

2. If/Else - Generally good, although you need to state that a decision does not have to be binary, it can have more than 2 possibilities 4/5

3. Loops - You haven't stated the difference between the two loops - for is when you know the number of iterations, while when you don't. You do allude to it in your examples however! Good examples shown. 4/5

PART B

Exercise 1

Logical Stages/Testing - Stages are stated, but you do not mention validation. Testing is presented OK, but you do not state whether the tests you performed caused the program to behave as expected. 3.5/5

Coding/Annotation - Good validation on car mile, but you have not validated the number of doors. Annotation is OK. 8/10

Exercise 2

Logical Stages/Testing - A couple of stages are a bit strange as there is no need to input pi or the formula, this can be hard-coded into the program. Same comments as above on testing. 7/10

Coding/Annotation - Inputs are well validated and annotations are OK. You could also consider the relative sizes of the triangle and the circle 8/10

Exercise 3

Logical Stages/Testing - A bit repetitive, you could have just said "prompt user to enter 5 prices"
Validation no mentioned (eg price cannot be negative) Again, no formulas are input for stages 23 and 24, you just calculate within the program.
Testing is OK, but you do not test for negative numbers. Same comment as above regarding showing the result. 7/10

Coding/Annotation - Generally good, but it would be a lot more efficient if you used a for loop to enter your products/prices. 11/15

Exercise 4 (Turtle)

Logical Stages/Testing - 8/10

Coding/Annotation - OK, but a bit repetitive. A function would be useful here (ie one function for N and another for O.) You could also take advantage of the rotational symmetry of the "N" to reduce the code needed. Good to see a loop for the "O" though! 9/15

TOTAL: 72%

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12
