

Assignment 1, Michael davies

by Michael Davies

Submission date: 10-Jan-2023 02:01AM (UTC+0000)

Submission ID: 194283573

File name: 42249_Michael_Davies_Assignment_1__Michael_davies_466019_169944784.docx (3.04M)

Word count: 1564

Character count: 7421

In programming, a variable is a storage location for a value. When you create a variable, you can give it a name and assign it a value, such as:

```
age = 27
name = "John"
```

Variables are important in programming because they allow you to store and manipulate data within your programs. You can use variables to store values such as numbers, strings, and other data types, and then perform operations on them or use them to control the flow of your program.

There are several different types of variables that you can use in Python, including:

Integer variables, which store whole numbers

Float variables, which store decimal numbers

String variables, which store sequences of characters

List variables, which store collections of other values[1]

If/elif/else statements are useful for making decisions in a program based on certain conditions. For example, you might want to check if a user has entered a valid password before allowing them to log in to an application, or you might want to check if a number is even or odd before performing a certain operation on it. [2]

```
x = 15
if x > 10:
    print("x is greater than 10")
elif x > 5:
    print("x is greater than 5 but not greater than 10")
else:
    print("x is not greater than 5")
```

Loops are used in programming to execute a block of code multiple times until a certain condition is met. This is useful when you want to perform a certain task repeatedly, such as reading data from a file or iterating over elements in a list. There are two types of loops in Python: for loops and while loops. A for loop is used to iterate over a sequence (such as a list or tuple) or other iterable objects. Here is an example of a for loop in Python:

```
fruits = ['apple', 'banana', 'mango']
for fruit in fruits:
    print(fruit)
```

This for loop will iterate over the list of the fruit and print each element in the list. The output will be:

```
apple
banana
mango
```

A while loop is used to repeat a block of code as long as a certain condition is true. Here is an example of a while loop in Python:

```
count = 0
while count < 5:
    print(count)
    count += 1
```

This while loop will continue to execute as long as the value of count is less than 5. The output will be:

```
0
```

1
2
3
4

It's important to note that if the condition in a while loop is always true, the loop will run forever, resulting in an infinite loop. This can be useful in some cases, but it's important to make sure that the loop will eventually terminate to avoid unintended consequences.[3]

The 1st step when writing code to gather the information from the user is to ask for their name and validate that it is a reasonable name not too long I used 15 characters since the average person's name in English is 7/8 characters this allowed me to have some room for people with larger names I done this for the surname as well once this was done I asked for the NI number I validated this as well making sure the length was 9 characters long once this was done I moved onto the age I validated this by making the sure have to be a minimum of 18 and a maximum of 120 then I had the user enter their wage which wasn't validated due to there being a very large amount of wage variety across the world so it allowed people from all over the world use it once this was don't it printed in all out at the bottom in an orderly fashion

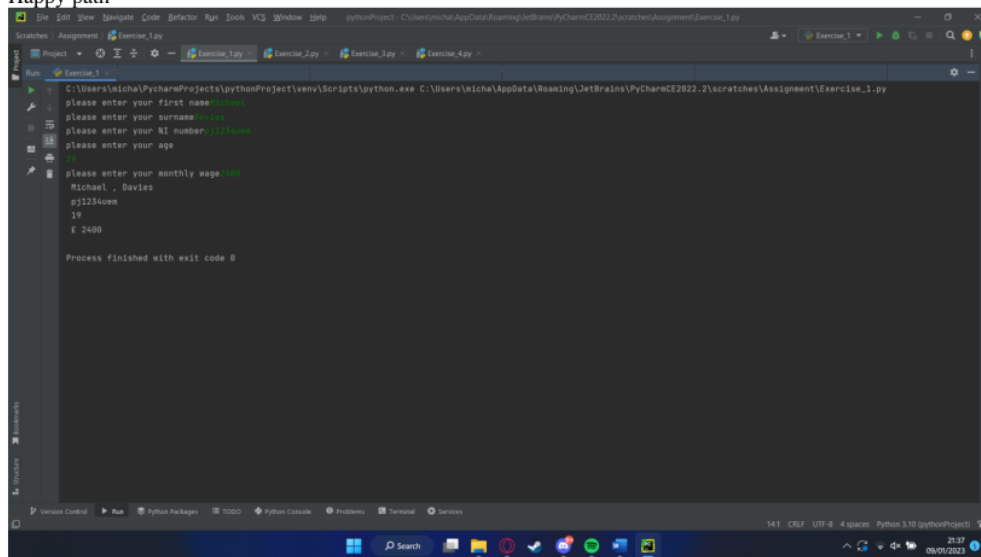
```
Name = input("please enter your first name")
while len(Name) >= 15:
    print("your name cannot be more than 15 characters please
reenter your 1st name")
    Name = input()
Name2 = input("please enter your surname")
while len(Name2) >= 15:
    print("your surname cannot be more than 15 characters please
reenter your surname")
    Name2 = input()

NI = input("please enter your NI number")
while len(NI) != 9:
    print("Are you sure this is your NI number it must be 2 letters
6 numbers and then 1 letter")
    print("please reenter your NI number")
    NI = input()

print("please enter your age")
age = int(input())
while 18 > age < 120:
    print("you are not over 18")
    print("please try again")
    age = int(input())
wage = int(input("please enter your monthly wage"))

print("", Name, ",", Name2, "\n", NI, "\n", age, "\n", "£", wage)
```

Happy path



```
pythonProject - C:\Users\micha\AppData\Roaming\JetBrains\PyCharmCE2022.2\scratches\Assignment\Exercise_1.py
Scratches / Assignment / Exercise_1.py
Run - Exercise_1.py
C:\Users\micha\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\micha\AppData\Roaming\JetBrains\PyCharmCE2022.2\scratches\Assignment\Exercise_1.py
please enter your first name Michael
please enter your surname Davies
please enter your NI number p1234567
please enter your age 19
please enter your monthly wage £ 2400
Michael , Davies
p1234567
19
£ 2400
Process finished with exit code 0
```

Minimum / max testing

The image displays two screenshots of a Python IDE, likely Visual Studio Code, showing the execution of a program that prompts the user for personal information and validates the input.

Top Screenshot: The program prompts the user for their first name, surname, NI number, and age. The user enters "1" for age. The program finishes with exit code 0.

```
pythonProject C:\Users\nicha\AppData\Roaming\JetBrains\PyCharmCE2022.2\scratches\Assignment\Exercise_1.py
Run Exercise_1
C:\Users\nicha\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\nicha\AppData\Roaming\JetBrains\PyCharmCE2022.2\scratches\Assignment\Exercise_1.py
please enter your first name
please enter your surname
please enter your NI number
Are you sure this is your NI number it must be 2 letters & numbers and then 1 letter
please reenter your NI number
Are you sure this is your NI number it must be 2 letters & numbers and then 1 letter
please reenter your NI number
please enter your age
1
please enter your monthly wage
123456789
120
E 1
Process finished with exit code 0
```

Bottom Screenshot: The program prompts the user for their first name, surname, NI number, and age. The user enters "12345678901234" for age. The program finishes with exit code 0.

```
pythonProject C:\Users\nicha\AppData\Roaming\JetBrains\PyCharmCE2022.2\scratches\Assignment\Exercise_1.py
Run Exercise_1
C:\Users\nicha\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\nicha\AppData\Roaming\JetBrains\PyCharmCE2022.2\scratches\Assignment\Exercise_1.py
please enter your first name
your name cannot be more than 15 characters please reenter your 1st name
please enter your surname
your surname cannot be more than 15 characters please reenter your surname
please enter your NI number
Are you sure this is your NI number it must be 2 letters & numbers and then 1 letter
please reenter your NI number
Are you sure this is your NI number it must be 2 letters & numbers and then 1 letter
please reenter your NI number
please enter your age
12345678901234
123456789
120
E 9999999999
Process finished with exit code 0
```

Major error / intentional not working

The image shows two screenshots of a Python IDE (likely VS Code) running a program. The program prompts the user for their first name, surname, NI number, and age. The first screenshot shows the program running successfully after the user enters 'John', 'Doe', '12345678901', and '25'. The second screenshot shows the program crashing with a `ValueError: invalid literal for int() with base 10: 'seventy'` when the user enters 'seventy' for their age. The error message is displayed in the console, indicating that the program is not handling non-numeric input correctly.`

```
File "C:\Users\nicha\PycharmProjects\pythonProject\venv\Scripts\python.exe", line 28, in <module>
    age = int(input("please enter your monthly wage"))
ValueError: invalid literal for int() with base 10: 'seventy'

Process finished with exit code 1
```

When calculating a square and a circle the diameter and width are needed so to start my code both are asked for the diameter is first halved to find the radius out and then the area of the circle is found out by doing pi times radius squared the area of the square is next which is Length squared once this is done validation happens to make sure the user has realistic figures and the square is smaller than the circle and then the area of the shaded part is

found out by taking the area of the circle and area of the square away from each other and then it prints the area shaded area out

```
# allow the user to enter the dimensions of the shapes
diameter = float(input('Enter the diameter of the circle: '))
width = float(input('Enter the length of the square: '))

# calculate the area of the circle
radius = diameter / 2
circle_area = 3.14 * radius * radius

# calculate the area of the square
square_area = width * width

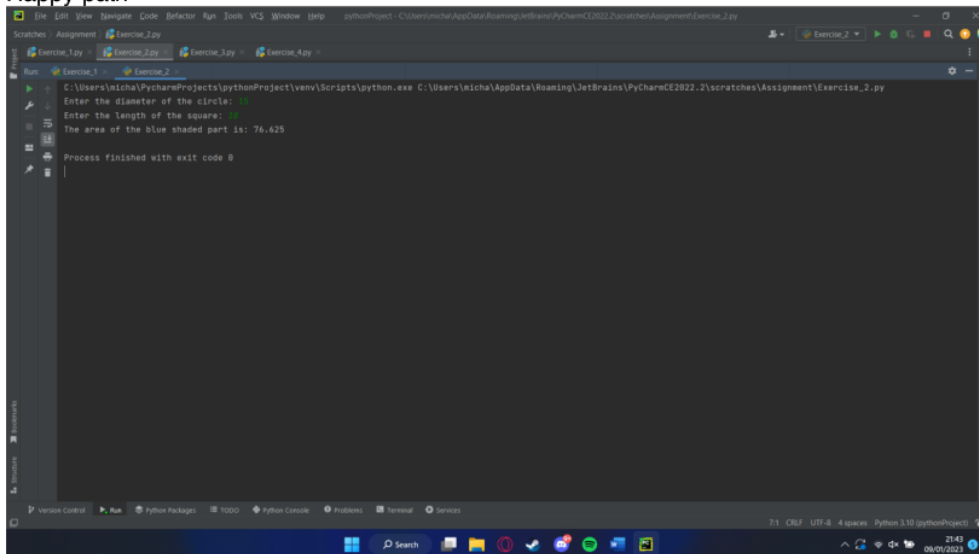
# extra credit part

# validate the user's input
if width > diameter:
    print('Error: The square cannot be larger than the circle.')
    exit()

# calculate the area of the blue shaded part
shaded_area = circle_area - square_area

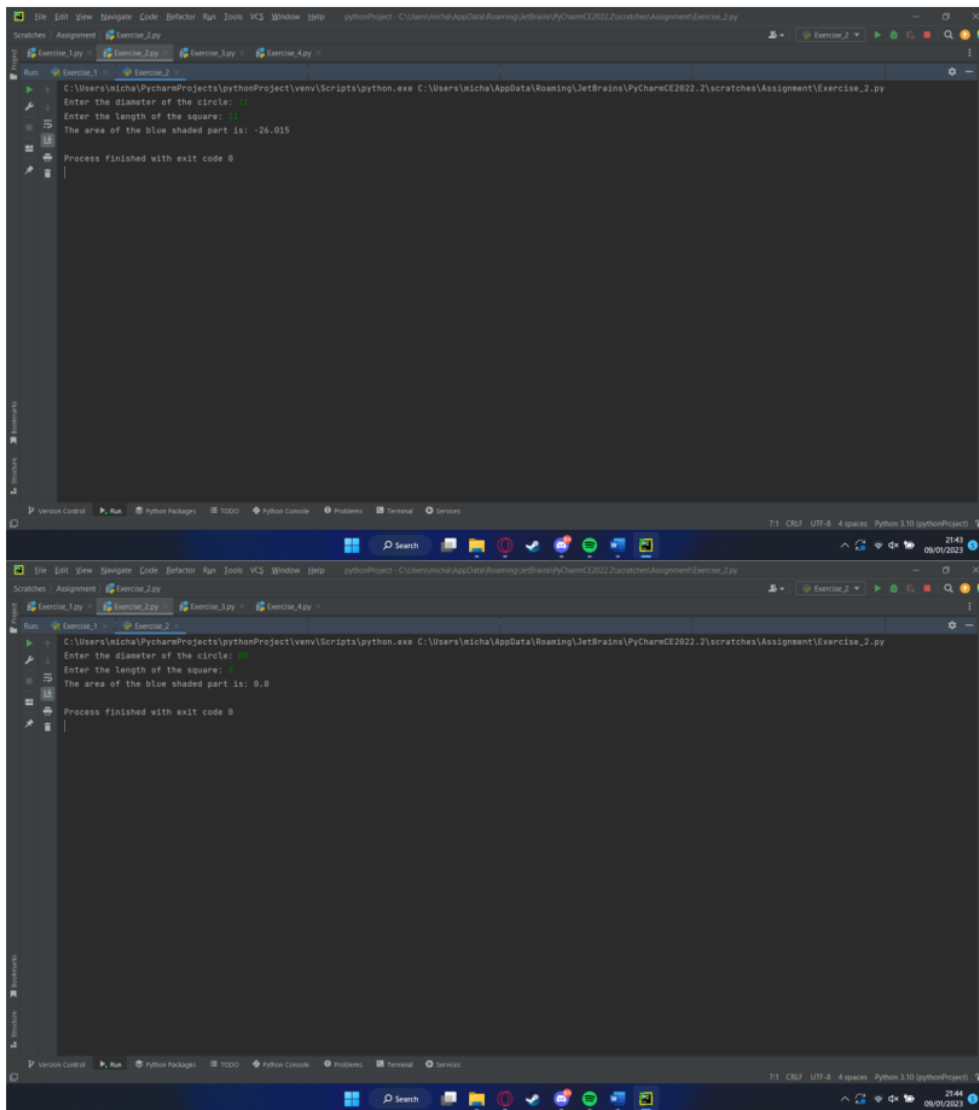
# print the area of the blue shaded part
print('The area of the blue shaded part is:', shaded_area)
```

Happy path



```
python C:\Users\nichal\AppData\Roaming\JetBrains\PyCharmCE2022.2\scratches\Assignment\Exercise_2.py
Enter the diameter of the circle: 10
Enter the length of the square: 5
The area of the blue shaded part is: 76.625
Process finished with exit code 0
```

Min/max



The screenshot shows a Python IDE window titled 'pythonProject - C:\Users\micha\AppData\Roaming\JetBrains\PyCharmCE2022.2\scratches\Assignment\Exercise_2.py'. The code in the editor is as follows:

```
Exercise_2.py
Enter the diameter of the circle: 10
Enter the length of the square: 5
The area of the blue shaded part is: 9.6915802469136+111
Process finished with exit code 0
```

The output window shows the same text as the code, indicating successful execution. The status bar at the bottom indicates '7:11 CRLF UTF-8 4 spaces Python 3.10 (pythonProject)'.

Breaks code

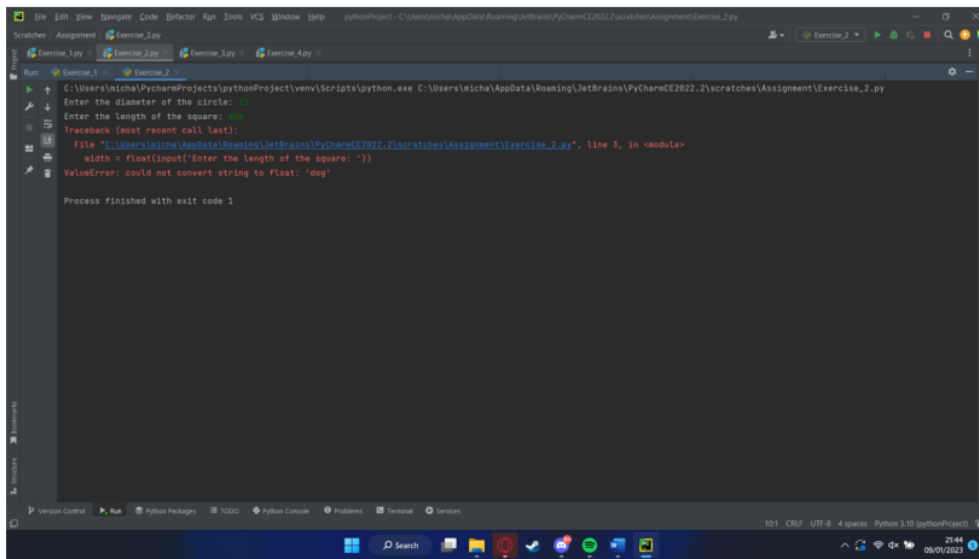
The screenshot shows the same Python IDE window, but now with a runtime error. The code is identical to the previous screenshot:

```
Exercise_2.py
Enter the diameter of the circle: 10
Enter the length of the square: 5
The area of the blue shaded part is: 9.6915802469136+111
Process finished with exit code 1
```

The output window displays a traceback error:

```
Traceback (most recent call last):
  File "C:\Users\micha\AppData\Roaming\JetBrains\PyCharmCE2022.2\scratches\Assignment\Exercise_2.py", line 2, in <module>
    diameter = float(input('Enter the diameter of the circle: '))
ValueError: could not convert string to float: 'cat'
```

The status bar at the bottom indicates '9:1 CRLF UTF-8 4 spaces Python 3.10 (pythonProject)'.



When doing a shopping list you need a list of course so the first thing I did was make a list and import random which would allow me to do something a bit later on once the list is made I made it so the user could put what items they want in the list once they are done they type "done" this then exits the list and says everything in the list and asked for the price of each item so it can be calculated once this is done the user is entered into a raffle where a discount is taken off the total this why the random was imported so it generates a raffle number then gets an amount to discount the shop by then applies the discount and then it gives the user the new amount

```
import random

# list for shopping
shopping_list = []

# Get the customer's shopping list
while True:
    item = input("Enter an item for your shopping list once finished type done: ")

    # Stop getting items when the customer enters 'done'
    if item.lower() == "done":
        break

    shopping_list.append(item)

# Print the shopping list
print("Your shopping list:")
```

```

for item in shopping_list:
    print(item)

# Calculate the total of the shopping list
total = 0
for item in shopping_list:
    price = float(input("Enter the price of " + item + ": "))
    total += price

# Print the total
print("Total: £" + str(total))

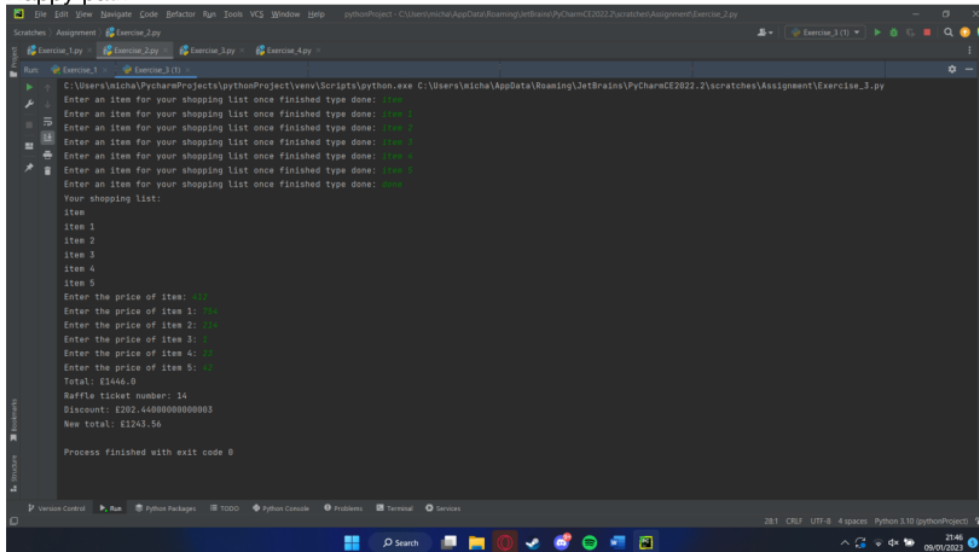
# Generate a random raffle ticket number between 10 and 40
ticket_number = random.randint(10, 40)

# Calculate the discount and new total
discount = ticket_number / 100 * total
new_total = total - discount

# Print the discount and new total
print("Raffle ticket number: " + str(ticket_number))
print("Discount: £" + str(discount))
print("New total: £" + str(new_total))

```

Happy path

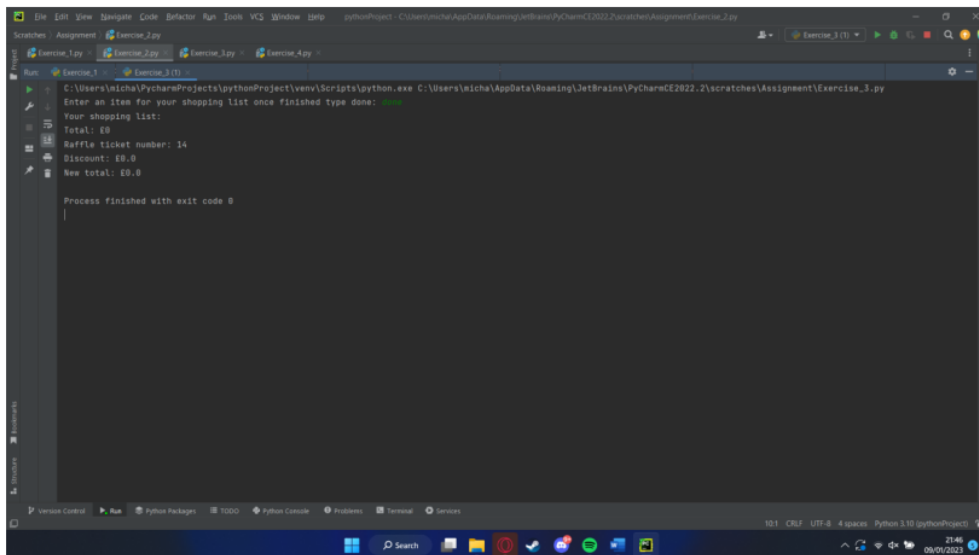


```

C:\Users\leicha\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\leicha\AppData\Roaming\JetBrains\PyCharmCE2022.2\scratches\Assignment\Exercise_3.py
Enter an item for your shopping list once finished type done: item
Enter an item for your shopping list once finished type done: item
Enter an item for your shopping list once finished type done: item
Enter an item for your shopping list once finished type done: item
Enter an item for your shopping list once finished type done: item
Enter an item for your shopping list once finished type done: item
Your shopping list:
item
item 1
item 2
item 3
item 4
item 5
Enter the price of item 1: 100
Enter the price of item 2: 100
Enter the price of item 3: 100
Enter the price of item 4: 100
Enter the price of item 5: 100
Total: £1444.0
Raffle ticket number: 14
Discount: £202.44000000000003
New total: £1241.56
Process finished with exit code 0

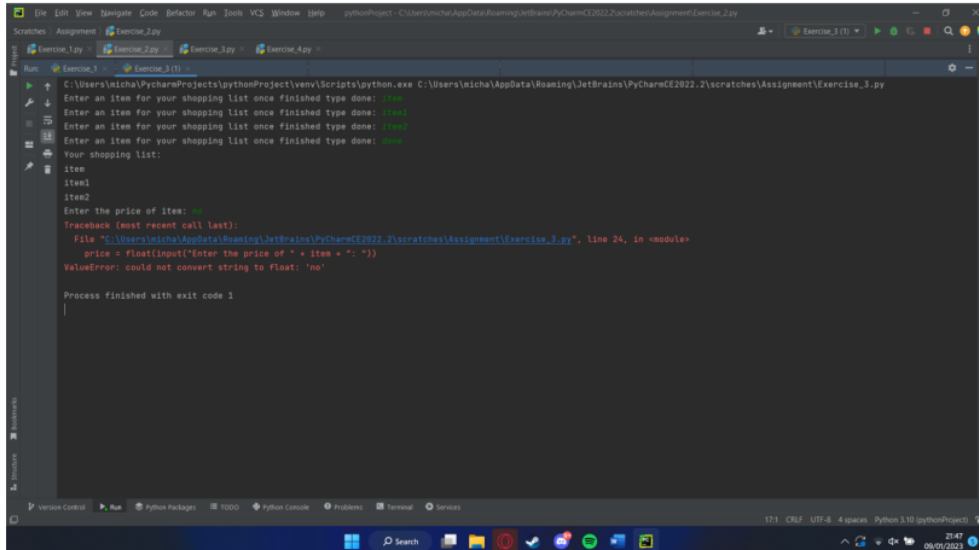
```

Min/max



```
File Edit View Navigate Code Refactor Run Tools VCS Window Help pythonProject - C:\Users\nicha\AppData\Roaming\JetBrains\PyCharmCE2022.2\scratches\Assignment\Exercise_3.py
Scratches Assignment Exercise_3.py Exercise_3.py Exercise_4.py
Run Exercise_3.py Exercise_3 (1)
C:\Users\nicha\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\nicha\AppData\Roaming\JetBrains\PyCharmCE2022.2\scratches\Assignment\Exercise_3.py
Enter an item for your shopping list once finished type done:
Your shopping list:
Total: £0
Raffle ticket number: 14
Discount: £0.0
New total: £0.0
Process finished with exit code 0
```

Breaks code



```
File Edit View Navigate Code Refactor Run Tools VCS Window Help pythonProject - C:\Users\nicha\AppData\Roaming\JetBrains\PyCharmCE2022.2\scratches\Assignment\Exercise_3.py
Scratches Assignment Exercise_3.py Exercise_3.py Exercise_4.py
Run Exercise_3.py Exercise_3 (1)
C:\Users\nicha\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\nicha\AppData\Roaming\JetBrains\PyCharmCE2022.2\scratches\Assignment\Exercise_3.py
Enter an item for your shopping list once finished type done:
Enter an item for your shopping list once finished type done:
Enter an item for your shopping list once finished type done:
Your shopping list:
item
item1
item2
Enter the price of item:
Traceback (most recent call last):
  File "C:\Users\nicha\AppData\Roaming\JetBrains\PyCharmCE2022.2\scratches\Assignment\Exercise_3.py", line 24, in <module>
    price = float(input("Enter the price of " + item + ": "))
ValueError: could not convert string to float: 'no'
Process finished with exit code 1
```

The first thing needed to draw the word “ZOO” in turtle is to import the turtle once that is done then you need to set a name for the turtle I chose “T” I also set the speed of the turtle and then the size I wanted it to draw in once that was all done I lifted the turtle up so it didn’t draw using t.penup() and then moving it to the starting position of the Z and then drew the “Z” once the “Z” was completed then it moved to the O using the pen up command again and then putting the location it should go to and drawing an O and then moving to the 2nd O and drawing that as well

```
import turtle
t = turtle.Turtle()
t.speed(0)
t.pensize(1)

# Move the turtle to the starting position for the "Z"
t.penup()
t.goto(-150, 100)
t.pendown()

# Draw the "Z"
t.forward(100)
t.right(135)
t.forward(150)
t.left(135)
t.forward(100)
t.right(90)
t.forward(10)
t.right(90)
t.forward(125)
t.right(135)
t.forward(150)
t.left(135)
t.forward(75)
t.right(90)
t.forward(10)
t.right(90)

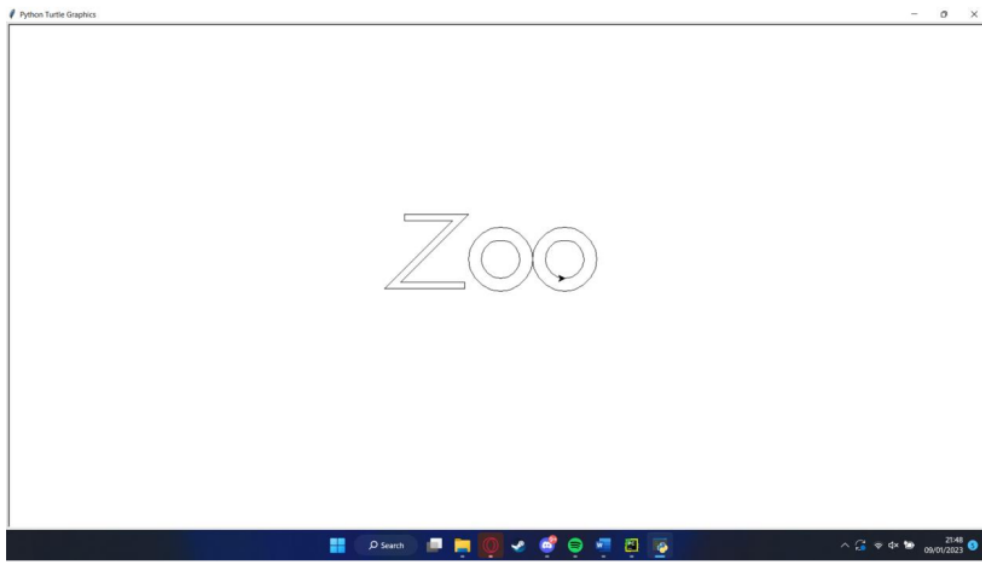
# Move the turtle to the starting position for the "O"
t.penup()
t.goto(0, -20)
t.pendown()

# Draw the first "O"
t.circle(50)
t.penup()
t.left(90)
t.forward(20)
t.right(90)
t.pendown()
t.circle(30)

# Move the turtle to the starting position for the second "O"
t.penup()
t.goto(100, -20)
t.pendown()

# Draw the second "O"
t.circle(50)
t.penup()
t.left(90)
t.forward(20)
t.right(90)
t.pendown()
t.circle(30)

turtle.done()
```



References

- [1] Stef Maruch, Aahz Maruch , Python For Dummies, Wiley, 2011.
- [2] "Understanding Python If-Else Statement," Simplylearn, 23 August 2022. [Online]. Available: <https://www.simplilearn.com/tutorials/python-tutorial/python-if-else-statement#:~:text=The%20if-else%20statement%20is,else%20block%20code%20is%20executed..> [Accessed 09 January 2023].
- [3] "Pyhton loops," W3school, [Online]. Available: https://www.w3schools.com/python/python_for_loops.asp. [Accessed 09 January 2023].

Assignment 1, Michael davies

GRADEMARK REPORT

FINAL GRADE

68 / 100

GENERAL COMMENTS

Instructor

PART A

A1 Variables - You have talked about variables and types, but you haven't talked about dynamic typing you need say specifically that the type is assigned (inferred) in Python based on the value entered. You need a bit more on this and also that in Python, types can change within the lifetime of the program

3 / 5

A2 If/Else - Explanation is OK, you could also have shown a multiple condition (and/or)

4 / 5

A3 Loops - A good explanation. Be careful with your example for the while loop, in that situation a for loop would be more appropriate because you know the number of iterations in advance (ie 0 to 4)

4/5

PART B

B1 Variables

Logical Stages/Testing - As a general comment I don't think you have consulted the accompanying document "How to lay out Part B of your Assignment"? - it's on Moodle and I also went through it in class. These are not really logical stages - your text reads more like a narrative of the code produced. The logical stages should show logic only and not refer to code. Testing is

OK but it is difficult to read as it's so small on the screenshot. Crop out just the parts you need for better presentation.

3/5

Code - Generally good, although you could have validated wage to not be a minus number (nowhere in the world pays in negative numbers!)

9/10

B2 Shape

Logical Stages/Testing - Same comment as above for logical stages. Testing is OK but could be a bit more extensive (see my comments about the code below)

2/5

Code - A reasonable answer. You need to validate so that the inputs for radius and diameter are not negative. The algorithm for the relative sizes of the square and circle is a bit lacking - the dimensions of the square do not have to be bigger than the diameter to exceed the circle's footprint.

9/15

B3 Lists

Logical Stages/Testing - Same comments as above for logical stages. Testing could be a bit more extensive (see Code comments below)

5/10

Code - A solution has been made - it would be better to store the prices in a list though so that the program has a record of them (in your code you overwrite them after they are entered) You could also have validated that the prices are not negative.

B4 Turtle

Code - A solution has been found, For efficiency, you could use a loop or a function for the O to avoid repeating code. Also, the Z has rotational symmetry so could also be looped.

General comments:

It is better for presentation if you put a title page on your work containing your name, course, module code/title and title of the assignment. Also, please number the questions to make it easier for the reader! It is also good practice to number the pages.

Avoid the use of "I" , "you", "we" etc in academic reports - this is the active voice and you need to use the passive voice. For example "I had the user enter their wage" -> "The user was prompted to enter their age".

OVERALL TOTAL FOR MODULE: 68%

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
