

Assignment 1

by Jae Mears

Submission date: 12-Jan-2023 02:32PM (UTC+0000)

Submission ID: 194724985

File name: 40189_Jae_Mears_Assignment_1_466019_494891358.docx (402.22K)

Word count: 932

Character count: 4820

Introduction to computer programming

A1.

A variable is a vessel for storing data values in code. Python is one of only languages that has no command for declaring a variable. There are a couple of types of variables for python for example numbers, strings and lists. These are the few we have learnt in class with depth. Variables help in applying the programming logic to variety your inputs in the code. In class we have used numbers, strings and lists, this is a important topic we have to learn because without them it's a hassle to try and code without the understanding of these variables. Python allows you to name is variable to own will and where you want to put it only if it works. They're important in program writing because they label the data, so you understand clearly by reading the code.

If and else statements are used to execute true parts from false parts in the program. If it's true the if statement will be executed in the code and if it is false the else statement will be executed in the same way.

```
fruits.pop()
del fruits[2]
del fruits[2]
fruits.remove("pear")

print("What is your favourite fruit, I'll see if i have it!")
userFruit = input()

if userFruit in fruits:
    print("Yes we have it")
else:
    print("sold out")
```

So, in this code example and you can see the way I use the "if" statement in this section of code. This allows the person using the program to enter their own fruit and the if statement pretty much says if we have the fruit available or not. If we don't have the fruit available the "else" statement comes in and says

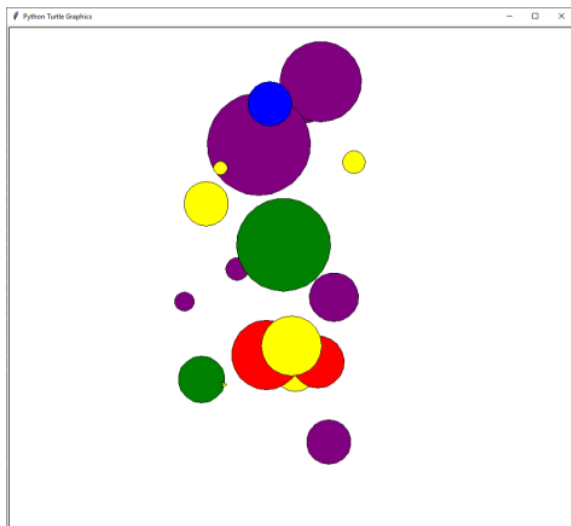
we're "sold out", this happens when the customer enters a fruit that we don't have in stock.

```
colors = ["Red", "Purple", "Yellow", "Pink", "Blue", "Green"]
myRandom3 = random.randint(5, 100)
print(len(colors))
for counter in range(0, myRandom3, 1):
    myRandom4 = random.randint(0, len(colors)-1)
    myTurt.fillcolor(colors[myRandom4])

    myRandom = random.randint(10, 100)
    myRandom1 = random.randint(-200, 200)
    myRandom2 = random.randint(-400, 400)

    myTurt.begin_fill()
    myTurt.penup()
    myTurt.goto(myRandom1, myRandom2)
    myTurt.pendown()
    myTurt.circle(myRandom)
    myTurt.end_fill()
turtle.done()
```

In this section of code, I produced a drawing or drawings or circles that can positions and colours each time a circle is drawn. This is why I used a loop because without a loop here it will be near impossible or very hard to produce a program like this without one. So, I need a loop here and I used "for counter in range(0, myRandom3, 1):" for this question and it worked out great.

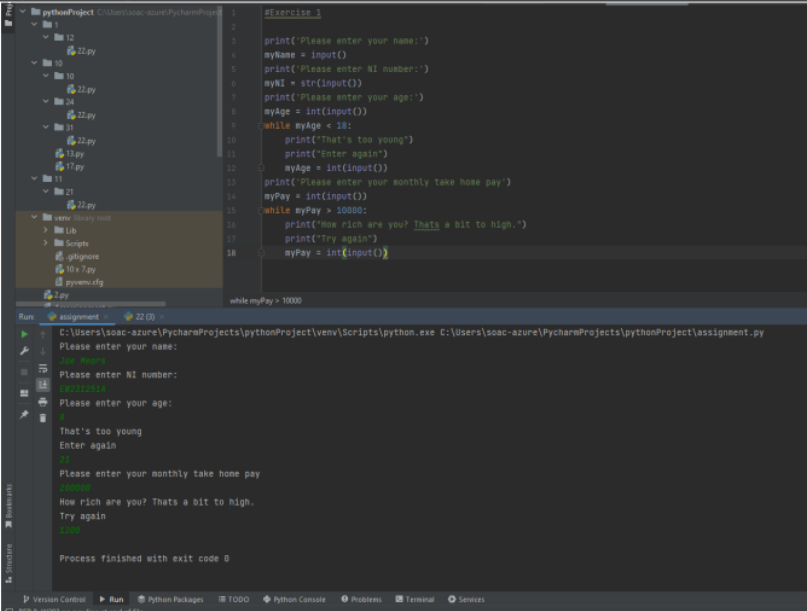


A2.

Exercise 1

I used my basic understanding to start off a template so I can practice answering the question before starting it so like a draft. After that I flew in and answered the question, I just followed the template of the example showed and based my code off that. These are a very basic understanding of code that allows the user using the program to enter in detail. In some of the questions I put a “while” statement in so for under 18 year olds I had a print statement that said “That’s too young”.

```
print('Please enter your name:')
myName = input()
print('Please enter NI number:')
myNI = str(input())
print('Please enter your age:')
myAge = int(input())
while myAge < 18:
    print("That's too young")
    print("Enter again")
    myAge = int(input())
print('Please enter your monthly take home pay')
myPay = int(input())
while myPay > 10000:
    print("How rich are you? That's a bit to high.")
    print("Try again")
    myPay = int(input())
```



The screenshot shows a Python IDE with a file explorer on the left, a code editor in the center, and a terminal at the bottom. The code editor displays the same code as the block above. The terminal shows the output of the program, including prompts for name, NI number, age, and monthly take home pay, and the corresponding user input and program responses.

```
Run: assignment - 22 (3)
C:\Users\soac-azure\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\soac-azure\PycharmProjects\pythonProject\assignment.py
Please enter your name:
Please enter NI number:
Please enter your age:
That's too young
Enter again
Please enter your monthly take home pay
How rich are you? That's a bit to high.
Try again
Process finished with exit code 0
```

Exercise 2

This allows you to put in the number of the square and it works it out for you.

```
def calculate_area(length, width):
    # Area of the outer rectangle
    outer_rectangle_area = length * width
    # Area of the inner square
    inner_square_area = (length - 2) * (width - 2)
    # Area of the blue shaded part
    blue_shaded_area = outer_rectangle_area - inner_square_area

    return blue_shaded_area

length = int(input("Enter the length of the shape: "))
width = int(input("Enter the width of the shape: "))

print("The area of the blue shaded part is:", calculate_area(length, width))
```

Exercise 3

So I made a list that loops with the print "Enter item (or 'done' to finish):" this allows the user of the program to continue to put their item in and when finished they can end the program and get the 10 to 40 percent discount and get the final price.

```
import random

# Initialize an empty shopping list
shopping_list = []

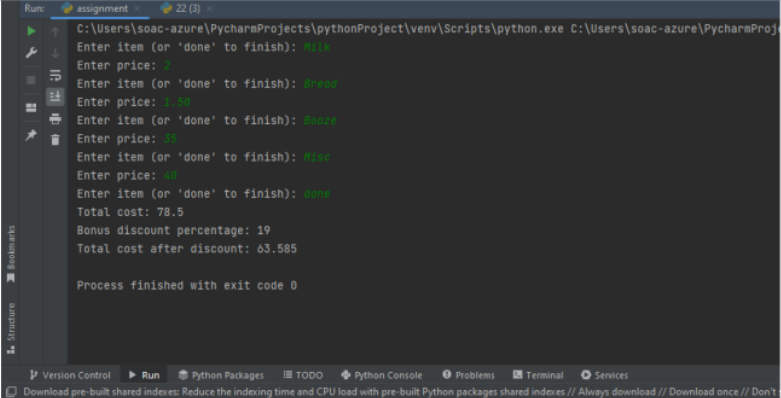
# Ask the customer to enter items and their prices
while True:
    item = input("Enter item (or 'done' to finish): ")
    if item.lower() == "done":
        break
    price = float(input("Enter price: "))
    shopping_list.append((item, price))

# Calculate the total cost of the shopping list
total = sum(item[1] for item in shopping_list)
print("Total cost:", total)

# Generate a random bonus discount percentage
discount_percent = random.randint(10, 40)
```

```
print("Bonus discount percentage:", discount_percent)

# Calculate the total cost after discount
total_after_discount = total - (total * (discount_percent / 100))
print("Total cost after discount:", total_after_discount)
```



Exercise 4

```
import turtle

# create turtle object
t = turtle.Turtle()

# set turtle's font size
t.pensize(5)

# set turtle's starting position
t.penup()
t.goto(-70, 0)
t.pendown()

# draw the letter "Z"
t.forward(50)
t.right(90)
t.forward(30)
```

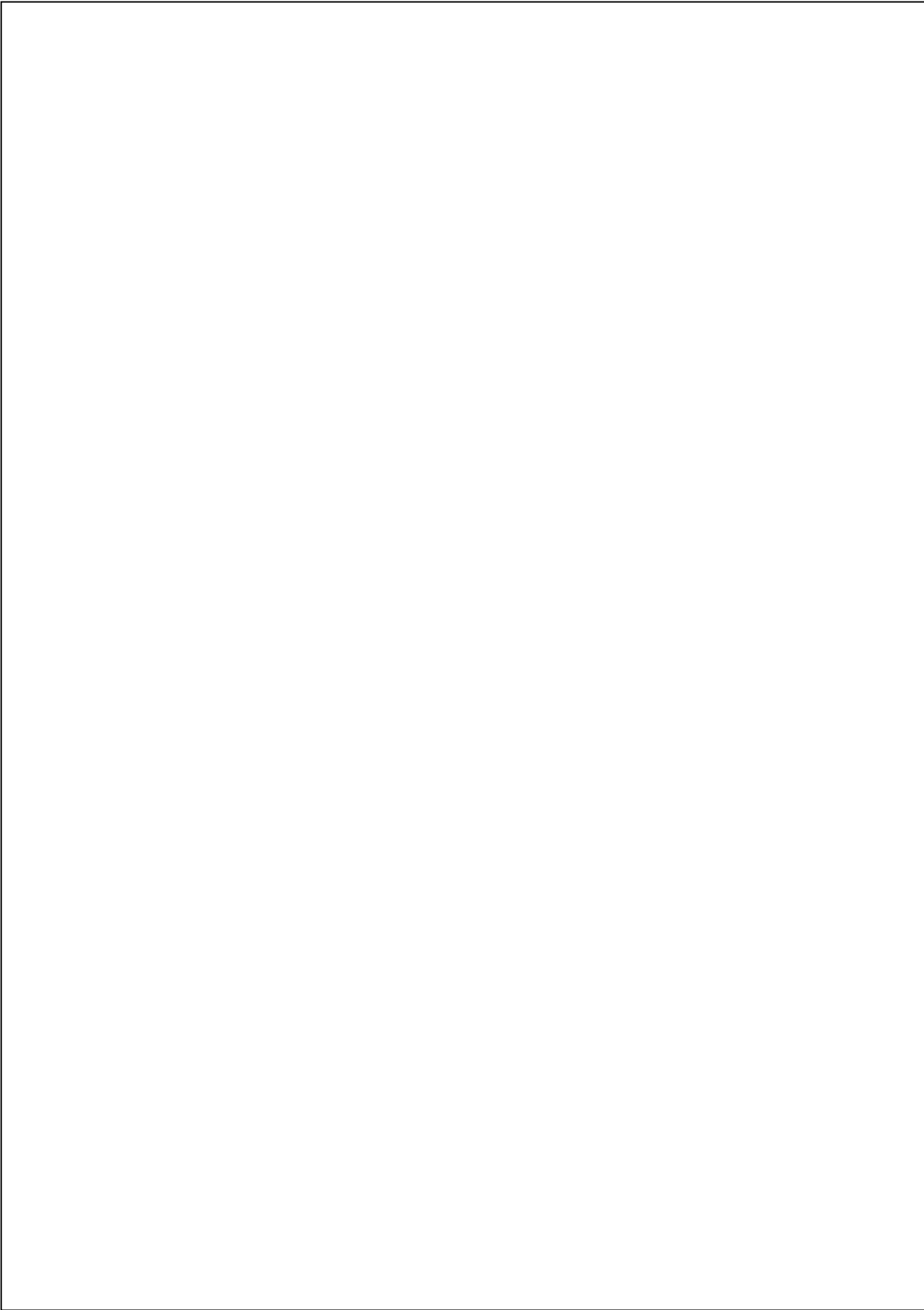
```

t.right(90)
t.forward(50)
t.left(90)
t.forward(30)
t.left(90)
t.forward(50)
# move turtle to next letter's starting position
t.penup()
t.goto(-30, 0)
t.pendown()
# draw the letter "0"
t.circle(15)
# move turtle to next letter's starting position
t.penup()
t.goto(10, 0)
t.pendown()
# draw the letter "0" again
t.circle(15)
# hide turtle
t.hideturtle()
# keep turtle window open
turtle.mainloop()

```



I did my best in coding the “zoo” in a 3d type of way but couldn’t get it to work out. This is my best try I had for this task.



Assignment 1

GRADEMARK REPORT

FINAL GRADE

53 / 100

GENERAL COMMENTS

Instructor

PART A

A1 Variables - You need to be a bit more precise. The types we have learned are int, float and 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)

2/5

A2 If/Else - Generally good, but you could have a bit more detail eg use of multiple conditions.

3 / 5

A3 Loops - You've said what a loop is, but you need to state the different types (for and while) and that for loops are counted, so the number of iterations is known before the loop starts (as opposed to a while loop where it is not known.) For loop example code is OK

2.5/5

PART B

B1 Variables

Logical Stages/Testing -

Logical Stages/Testing - 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. See document "How to Lay out assignment Part B" on Moodle. Testing is OK, but don't forget to use a wide range of values. It's also a bit small, crop your screenshot to just to part needed

3/5

Code - Generally good, you could also have validated for a minus number on myPay.

8/10

B2 Shape

Logical Stages/Testing - Same comment as above. Also, no testing provided 0/5

Code - A bit confusing, you don't seem to have done anything with the circle? Also, you need to validate the input to prevent the user entering minus numbers.

6/15

B3 Lists

Logical Stages/Testing - Logical stages, same comment as above. Testing only shows the "happy path" you need to use a wider range of values.

4/10

Code - Generally good, but you could have validated for minus numbers on the price. I'm not sure why you have only used one list, it would be a lot easier with the items in one and the prices in the other. Nevertheless, a good attempt.

14/20

B4 Turtle

Code - Although it's not quite there, you've had a go. You could have made things more efficient by using a loop or function for the Os. Also, a 3D Z has rotational symmetry, so you could loop.

10/20

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. It is also good practice to number the pages to make it easier for the reader.

You should definitely not use "I" or "you" in an academic assignment. This is what we call the active voice, you need to use the passive voice. For example, "I need a loop here" -> "A loop is needed here". Try to write in a formal way, you're a bit colloquial at times.

Some good work here, be careful to read the assignment brief to make sure you have everything in.

OVERALL TOTAL FOR MODULE: 53%

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5
