

a1

by Peter Siow

Submission date: 29-Apr-2022 12:59AM (UTC+0100)

Submission ID: 177739521

File name: 33390_Peter_Siow_a1_410175_1133006456.docx (332.22K)

Word count: 3318

Character count: 15619



SOFTWARE DEVELOPMENT

Assignment 1 – Due 29/04/2022



Peter Siow

Student Number – 2104217

Course Title – Computer Networks and
Cybersecurity

Contents

Part A.....	2
Variables	2
What is a variable?.....	2
Why are variables important in writing programs?.....	2
If/Else statement.....	2
How are if/else statements used	2
Why are if/else statements used	2
Loops.....	3
Why loops are needed in a program	3
Part B.....	4
Exercise 1:	4
Logical Stages – Exercise 1	4
Test 1 – Exercise 1	6
Test 2 – Exercise 1	6
Test 3 – Exercise 1	6
Test 4 – Exercise 1	7
Exercise 2:	7
Logical Stages – Exercise 2	7
Test 1 – Exercise 2	9
Test 2 – Exercise 2	9
Test 3 – Exercise 2	10
Test 4 – Exercise 2	10
Test 5 – Exercise 2	11
Exercise 3:	11
Logical Stages – Exercise 3	11
Test 1 – Exercise 3	14
Test 2 – Exercise 3	15
Test 3 – Exercise 3	16
Exercise 4:	16

Part A

Variables

What is a variable?

A variable is a value which is created prior in writing the main body code as a user can create a variable providing it with a name and assigning a value towards the variable created. The user can cast a variable using the three variable types in which if the user specifies the type, it when inputting a value, a user will only be allowed to input the value which is casted, an example of this would be a variable being called "age" and then casted as an integer the user will only be able to input values which are whole numbers.

Why are variables important in writing programs?

Variables are important when writing programs because if a user is writing a large program, then the variables created allow the user to go back and use the variables in code which is used later in the program. As it is especially important to use variables whenever using them in a program which is required for calculations for a formula such as a volume of a circle.

If/Else statement

How are if/else statements used

If/else statements are used in programming when a user wants to find if a condition is true, it will print out a statement which identifies if the statement is true and if the condition is false in the it will print out a different output in which will be a different result. As this allows a program to provide more than one output to a user which can be specified to the user specifically.

Why are if/else statements used

If/else statements can be used in programming if a condition can output more than one which can be different in which an example can be for grading for exams, as in the code it has three different grading criteria in which a user can receive depending on the mark.

Example 1:

```
userMark = float(input("Enter your mark: "))
```

```
if userMark >= 70:
```

```
    print("That is a distinction")
```

```
elif userMark >= 60:
```

```
    print("That is a merit")
```

```
elif userMark >= 40:
```

```
    print("That is a Pass")
```

```
else:
```

```
    print("Please resit the paper")
```

Example 2:

```
Pizza = input("What is your favourite topping ")
```

```
if Pizza == "Pineapple":  
    print("Should not be on pizza")  
elif Pizza == "Ham":  
    print("Good choice")  
elif Pizza == "Bacon":  
    print("The best topping!")  
else:  
    print("I have seen better")
```

Example 3:

```
Number1 = int(input("Please enter a number"))  
if Number1 == 0:  
    print("Exactly 0")  
elif Number1 > 0:  
    print("That number is a positive")  
else:  
    print("That is a negative number")
```

Loops

Why loops are needed in a program

Loops are required in programming if a program is required looping techniques, as there are two distinct types of loops which a user can use in a program in which are for and while loops. A while is a loop in which a piece of code will be continuously executed until the condition is true. A for loop is useful to a user as the program can reiterate over what the user has input into the code and provides an output which can be viewed as a list.

While Loop example:

```
print("Please enter your age:")  
age = int(input())  
while age < 18:  
    print("You're not old enough to come in")  
    age = int(input("Please enter your age:"))  
  
print("Welcome in the Club, enjoy your time!")
```

For loop example:

```
myGroceries = ["Oats", "Apples", "Rice", "Ice cream", "Water", "Bacon"]
```

```
for x in myGroceries:
```

```
    print(x)
```

Part B

Exercise 1:

Firstly, I created prints which prompts the user to enter a value to the according code which appears on the screen when the program is running. Secondly, I assigned variable names for when the user enters a value which I added using the user input command in the code. Then, I used loops in which will avoid any problems with the values which a user enters when prompted on the program.

Example would be for the age code if the user would enter a negative number. Finally, I used print again to display all the values which the user entered in the program to be displayed once all values have been entered.

Logical Stages – Exercise 1

- 1) Prompts user to enter their name (If numbers appear in the team it will loop until user gives name with only letters)
- 2) Prompts user to enter their age (Only accepts value above 1 anything less than 1 is rejected and looped until the user meets the condition)
- 3) Prompts user to enter their height in metres (If user enters height greater than 1.93 or less than 1 it will give them a prompt to enter a valid height.)
- 4) Prints out the values which the user entered into name, age and height.

Code:

```
print("Please enter your name:") # prompts user to enter name
```

```
while True: # loops program until condition is true
```

```
    name = input()
```

```
    if not name.isalpha(): # checks if all the information being entered is letters rather than letters and numbers.
```

```
        print ("Numbers are not allowed")
```

```
    else:
```

```
        break
```

```
print("Please enter your age:") # prompts the user to enter their age
```

```
age = int(input()) # allows the user to enter the age float value is not allowed only whole numbers  
and no string values
```

```
while age < 1:
```

```
    print("Not a valid age")
```

```
    age = float(input("Please enter your age:"))
```

```
print("Please enter your height in metres: ") # prompts the user to enter their height
```

```
userHeight = float(input()) # casted to float meaning height can be determined by decimal point
```

```
while userHeight > 1.93 or userHeight < 1:
```

```
    # loop which if the user enters a height greater than 1.93 it will loop until the user will input a  
    height less than the value
```

```
    print("Enter Valid height!")
```

```
    userHeight = float(input("Please enter your height in metres: "))
```

```
print("Thanks, you've entered:", name, "-", age, "years old -", userHeight, "metres tall") # displays all  
the values which have been entered by the user.
```

Test 1 – Exercise 1

```
Please enter your name:
Peter
Please enter your age:
21
Please enter your height in metres:
1.7
Thanks, you've entered: Peter - 21 years old - 1.7 metres tall
> |
```

The first test is when a user would enter valid values which are sensible to the program. In which passes causing no errors.

Test 2 – Exercise 1

```
Please enter your name:
444
Numbers are not allowed
p3t3r
Numbers are not allowed
|
```

The second test does not allow a user to enter numbers as a value by themselves and as well if the user were to replace letters for numbers in their name, in which the program will continue to loop until the user enters a valid name.

Test 3 – Exercise 1

```
Please enter your name:
Peter
Please enter your age:
-2
Not a valid age
Please enter your age:|
```

This tests if a user were to enter a value lower than 1 as they are not valid ages in which once entered an age less than 1 the user is prompted to enter another value and it loops until condition is met.

Test 4 – Exercise 1

```
Please enter your name:
peter
Please enter your age:
21
Please enter your height in metres:
2
Enter Valid height!
Please enter your height in metres: 0
Enter Valid height!
Please enter your height in metres: 1.92
Thanks, you've entered: peter - 21 years old - 1.92 metres tall
> |
```

This tests when a user will type in a height if the height is greater than 1.92 or less than 1 then it will loop and prompt the user to enter a value which is in the scale between 1 and 1.92.

Exercise 2:

Firstly, I created prompts asking for the values of the radius, length, width, and size of the shapes. Secondly, implemented a user input which also contain casted values and variables which will be fetched in the program later for the calculations. Then I created while loops for any false numbers which could have been entered such as negative numbers, so this avoids any errors in the program. Finally, I had to use the variables for the beginning of the program to calculate the eventual answer for the shaded area and I had to think logically as the formulas were different and this allowed me to use all the user inputs together to find the shaded area and once I figured out the logic I created a print which displayed the eventual answer and made sure to round the answer to one decimal places for a rounded answer.

Logical Stages – Exercise 2

- 1) Prompts user to enter radius of the circle (anything less than 1 or greater than 100 is not a valid value)
- 2) Prompts the user to enter the side of square (values less than 1 and greater than 100 are not accepted and user will need to choose a value suitable to be able to continue)
- 3) Prompts the user to enter width of rectangle (values less than 1 and greater than 11 are not accepted and user will need to choose a value suitable to be able to continue)
- 4) Prompts user to enter length of rectangle (values less than 1 and greater than 100 are not accepted and user will need to choose a value suitable to be able to continue)
- 5) Calculate the area of the circle
- 6) Calculates area of square
- 7) Calculates the area of rectangle
- 8) Calculates the total area of square and rectangle
- 9) Calculates the shaded area by using the area of circle – total area
- 10) Prints out the last answer of the shaded area.

Code:

```
print("Please enter radius of circle") # prompts user to enter radius of the circle

userRadius = float(input()) # allows the user to enter a value and then stores the value into the
userRadius variable

while userRadius < 1 or userRadius > 100: # while loop until condition is found true
    print("Invalid value, please try again:")
    userRadius = float(input("Please enter radius of circle:"))

print("Please enter side of square:") # prompts user, allows a user to know what to type in
userSideSquare = float(input())

while userSideSquare < 1 or userSideSquare > 100: # while loop until condition is found true
    print("Invalid value, please try again:")
    userSideSquare = float(input("Please enter side of square:"))

print("Please enter width of rectangle:") # prompts user, allows a user to know what to type in
userWidthRectangle = float(input())

while userWidthRectangle > 11 or userWidthRectangle < 1: # while loop until condition is found true
    print("Invalid value, please try again:")
    userWidthRectangle = float(input("Please enter width of rectangle:"))

print("Please enter length of rectangle") # prompts user, allows a user to know what to type in
userLengthRectangle = float(input())

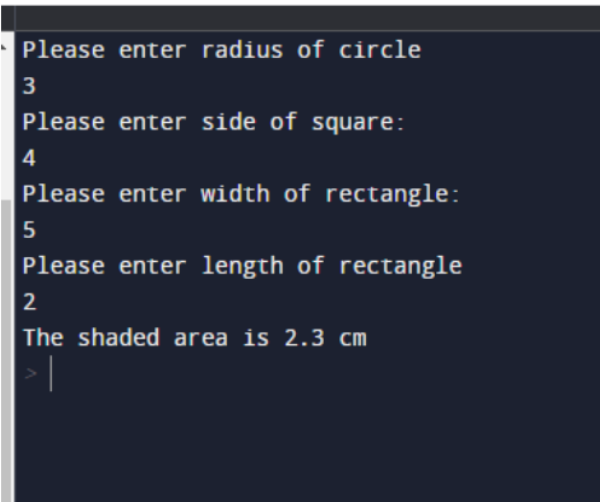
while userLengthRectangle < 1 or userLengthRectangle > 100 : # while loop until condition is found
true
    print("Invalid value, please try again:")
    userLengthRectangle = float(input("Please enter length of rectangle:"))
```

```
areaCircle = 3.14159 * (userRadius * userRadius) # calculates the area of the circle
areaSquare = userSideSquare * userSideSquare # calculates the area of a square
areaRectangle = userLengthRectangle * userWidthRectangle # calculates the area of rectangle
totalArea = areaSquare + areaRectangle # calculates the total area
```

```
shadedArea = areaCircle - totalArea # calculates the shaded area
```

```
print("The shaded area is", round (shadedArea, 1), "cm") # displays the final answer and rounds the
answer to one decimal place
```

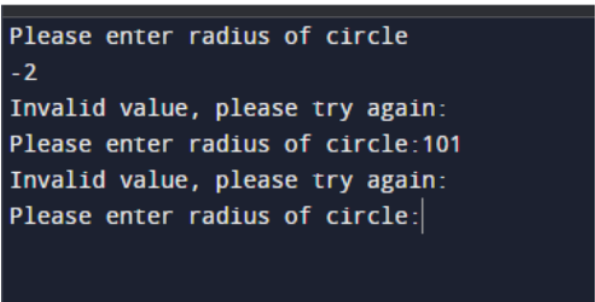
Test 1 – Exercise 2



```
Please enter radius of circle
3
Please enter side of square:
4
Please enter width of rectangle:
5
Please enter length of rectangle
2
The shaded area is 2.3 cm
> |
```

First test is values which are normal in the program which a user may enter for the calculation for the shaded area.

Test 2 – Exercise 2



```
Please enter radius of circle
-2
Invalid value, please try again:
Please enter radius of circle:101
Invalid value, please try again:
Please enter radius of circle:|
```

Second test is when a user would enter incorrect values in program for the radius with a negative number and a number greater than 100

Test 3 – Exercise 2

```
Please enter radius of circle
2
Please enter side of square:
-2
Invalid value, please try again:
Please enter side of square:101
Invalid value, please try again:
Please enter side of square:|
```

Third test is testing values which are negative and numbers greater than 100 for the side of square.

Test 4 – Exercise 2

```
^ Please enter radius of circle
2
Please enter side of square:
2
Please enter width of rectangle:
-2
Invalid value, please try again:
Please enter width of rectangle:101
Invalid value, please try again:
Please enter width of rectangle:|
```

This test is entering incorrect values which a user would input for the width of rectangle in which either can be a negative number or a number greater than 11.

Test 5 – Exercise 2

```
Please enter radius of circle
2
Please enter side of square:
2
Please enter width of rectangle:
2
Please enter length of rectangle
-2
Invalid value, please try again:
Please enter length of rectangle:1000
Invalid value, please try again:
Please enter length of rectangle:|
```

This test is when incorrect values have been entered in for the length of the triangle which is negative and number greater than 100.

Exercise 3:

Firstly, I created a list which included all the days in a week and the variables holds all the days and this is useful as I can fetch a day in the list rather than writing the day in the code each time, after I then added a prompt which ask the user to provide an input which is then stored in the variable and I did this for all the days, when writing the prompt instead of writing the days I would fetch them from the list created from the beginning. Secondly, I implemented while loops again for avoiding any issues of false information being added into the program while running. Finally, I had to create a calculation which displays the total mileage and petrol cost for the week in which for the total mileage I had to add all the days together of all the miles the user input into each prompt and then that provided me with the total mileage for the week and the petrol cost calculation was the total mileage divided by 11.2 and then multiplied by 1.35. As well for this I rounded the total mileage to 1 decimal place and the petrol cost to 2 decimal places.

Logical Stages – Exercise 3

1)List of all the days of the week

2)Prompts user to enter miles for Monday, by fetching the data store from the list and will store the value the user enters the list for "0" in the list (values less than 1 and greater than 500 are rejected and loop until user enters a value between 1 and 500)

3) Prompts user to enter miles for Tuesday, by fetching the data store from the list and will store the value the user enters the list for "1" in the list (values less than 1 and greater than 500 are rejected and loop until user enters a value between 1 and 500)

4) Prompts user to enter miles for Wednesday, by fetching the data store from the list and will store the value the user enters the list for "2" in the list (values less than 1 and greater than 500 are rejected and loop until user enters a value between 1 and 500)

5) Prompts user to enter miles for Thursday, by fetching the data store from the list and will store the value the user enters the list for "3" in the list (values less than 1 and greater than 500 are rejected and loop until user enters a value between 1 and 500)

6) Prompts user to enter miles for Friday, by fetching the data store from the list and will store the value the user enters the list for "4" in the list (values less than 1 and greater than 500 are rejected and loop until user enters a value between 1 and 500)

7) Prompts user to enter miles for Saturday, by fetching the data store from the list and will store the value the user enters the list for "5" in the list (values less than 1 and greater than 500 are rejected and loop until user enters a value between 1 and 500)

8) Prompts user to enter miles for Sunday, by fetching the data store from the list and will store the value the user enters the list for "6" in the list (values less than 1 and greater than 500 are rejected and loop until user enters a value between 1 and 500)

9) Calculates the total milage by adding all the days from the list together and prints out the total milage

10) Calculates the petrol cost by using the total mileage and the given values of £1.35 a litre and 11.2miles to the litre and prints out the cost of petrol.

Code:

```
daysOfWeek = ["Monday", "Tuesday", "Wedndesday", "Thursday", "Friday", "Saturday", "Sunday",] #  
list of all the days in a week
```

```
print("Enter miles for ", daysOfWeek[0])
```

```
daysOfWeek[0] = float(input()) # user input to enter number and then value is stored in the variable  
list
```

```
while daysOfWeek[0] < 1 or daysOfWeek[0] > 500: # while loop to avoid any errors in the program
```

```
    print("Please try again")
```

```
    print("Enter miles for Monday")
```

```
    daysOfWeek[0] = float(input())
```

```
print("Enter miles for ", daysOfWeek[1])
```

```
daysOfWeek[1] = float(input()) # user input to enter number and then value is stored in the variable  
list
```

```
while daysOfWeek[1] < 1 or daysOfWeek[1] > 500: # while loop to avoid any errors in the program
```

```
    print("Please try again")
```

```
    print("Enter miles for Tuesday")
```

```
    daysOfWeek[1] = float(input())
```

```
print("Enter miles for ", daysOfWeek[2])

daysOfWeek[2] = float(input()) # user input to enter number and then value is stored in the variable
list

while daysOfWeek[2] < 1 or daysOfWeek[2] > 500: # while loop to avoid any errors in the program

    print("Please try again")

    print("Enter miles for Wednesday")

    daysOfWeek[2] = float(input())


print("Enter miles for ", daysOfWeek[3])

daysOfWeek[3] = float(input()) # user input to enter number and then value is stored in the variable
list

while daysOfWeek[3] < 1 or daysOfWeek[3] > 500: # while loop to avoid any errors in the program

    print("Please try again")

    print("Enter miles for Thursday")

    daysOfWeek[3] = float(input())


print("Enter miles for ", daysOfWeek[4])

daysOfWeek[4] = float(input()) # user input to enter number and then value is stored in the variable
list

while daysOfWeek[4] < 1 or daysOfWeek[4] > 500: # while loop to avoid any errors in the program

    print("Please try again")

    print("Enter miles for Friday")

    daysOfWeek[4] = float(input())


print("Enter miles for ", daysOfWeek[5])

daysOfWeek[5] = float(input()) # user input to enter number and then value is stored in the variable
list

while daysOfWeek[5] < 1 or daysOfWeek[5] > 500: # while loop to avoid any errors in the program

    print("Please try again")

    print("Enter miles for Saturday")

    daysOfWeek[5] = float(input())
```

```

print("Enter miles for ", daysOfWeek[6])

daysOfWeek[6] = float(input()) # user input to enter number and then value is stored in the variable
list

while daysOfWeek[6] < 1 or daysOfWeek[6] > 500: # while loop to avoid any errors in the program

    print("Please try again")

    print("Enter miles for Sunday")

    daysOfWeek[6] = float(input())


totalMileage = daysOfWeek[0] + daysOfWeek[1] + daysOfWeek[2] + daysOfWeek[3] +
daysOfWeek[4] + daysOfWeek[5] + daysOfWeek[6] # calculation variable for the total mileage

print("The total mileage was:", round(totalMileage, 1)) # displays the total mileage and rounds up to
1 decimal place


petrolCost = totalMileage / 11.2 * 1.35 # calculation of the petrol cost

print("Cost of Petrol:",round (petrolCost, 2)) # displays the petrol cost and rounds up to 2 decimal
place

```

Test 1 – Exercise 3

```

Enter miles for  Monday
142
Enter miles for  Tuesday
23
Enter miles for  Wednesday
132
Enter miles for  Thursday
65
Enter miles for  Friday
43.2
Enter miles for  Saturday
323
Enter miles for  Sunday
53
The total mileage was: 781.2
Cost of Petrol: 94.16
> |

```

First test is with normal values entered into the program and in which displays all total mileage and cost of petrol for the week after all values have been entered.

Test 2 – Exercise 3

```
Enter miles for Monday
-3
Please try again
Enter miles for Monday
4
Enter miles for Tuesday
-2
Please try again
Enter miles for Tuesday
4
Enter miles for Wednesday
-2
Please try again
Enter miles for Wednesday
4
Enter miles for Thursday
-22
Please try again
Enter miles for Thursday
4
Enter miles for Friday
-33
Please try again
Enter miles for Friday
42
Enter miles for Saturday
-33
Please try again
Enter miles for Saturday
22
Enter miles for Sunday
-33
Please try again
Enter miles for Sunday
22.2
```

This test is when I entered negative numbers into each day and then followed up with a correct value when the program is running.

Test 3 – Exercise 3

```
Enter miles for Monday
666
Please try again
Enter miles for Monday
43
Enter miles for Tuesday
999
Please try again
Enter miles for Tuesday
54
Enter miles for Wednesday
999
Please try again
Enter miles for Wednesday
44
Enter miles for Thursday
999
Please try again
Enter miles for Thursday
454
Enter miles for Friday
999
Please try again
Enter miles for Friday
44
Enter miles for Saturday
999
Please try again
Enter miles for Saturday
44
Enter miles for Sunday
999
Please try again
Enter miles for Sunday
44
```

Test is with numbers greater than 500 then followed with correct values.

Exercise 4:

```
import random
```

```
animals = ["Dog", "Cat", "Rhino"]
```

```
results = random.choices(animals, weights=[2, 2, 2], k=2)
```

```
print("Welcome to the pairs game! There are 6 pairs to find")
```

```
print("Here is the board:")
```

```
print("0 1 2 3 4 5 6 7 8 9 10 11")
```

```
print("? ? ? ? ? ? ? ? ? ?")
```

```
print("Please enter guess 1:")
```

```
choice1= int(input())
```

```
print("Please enter guess 2:")
```

```
choice2 =int(input())
```

```
if choice1 == 0 and choice2 == 1:
```

```
    print(results)
```

```
elif choice1 == 1 and choice2 ==0:
```

```
    print(results)
```

```
elif choice1 == 1 and choice2 ==2:
```

```
    print(results)
```

```
elif choice1 == 2 and choice2 == 1:
```

```
    print(results)
```

```
#while results == animals[0] and animals[0]:
```

```
if results == (animals[0] and animals[0]):
```

```
    print("Match")
```

```
elif results == (animals[1] and animals[1]):
```

```
    print("Match")
```

```
elif results == (animals[2] and animals[2]):
```

```
    print("Match")
```

```
else:
```

```
    print("Not a match")
```

FINAL GRADE

55 / 100

GENERAL COMMENTS

Instructor**PART A**

Variables - You have talked about variables and alluded to types, but you need to say what the types are ie int, float string, you only mention int. Your explanation is a bit confused 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

If/Else - You've got the main points, could also have talked about and/or multiple conditions. 4/5

Loops - You haven't quite got the point about a for loop. Yes, they can be used with lists, but the important point is that the number of iterations is known before the program starts 4/5

PART B**Exercise 1**

Logical Stages-Testing - Logical stages generally good, Testing also OK, see comment about upper limit of age below 4.5 /5

Coding-Annotation - Generally good, could also have validated upper limit on age 9 /10

Exercise 2

Logical Stages-Testing - Good, but need to include formulae in logical stages. Testing OK but see comment about relative sizes below. 3.5 /5

Coding-Annotation - Working program produced, some validation of input range shape dimensions. You also need to consider the relative sizes of square, rectangle and circle - if the square/rectangle ends up bigger than the circle, the program will not work correctly. 7/10

Exercise 3

Logical Stages-Testing - Logical stages OK. Testing OK, could be a little wider 8 /10

Coding-Annotation - I'm a bit mystified that you have overwritten daysoftheweek - use a separate list for mileage. You need to loop as well as you have many repetitions of your code which leads to inefficiency. Some validation present 5.5 /10

Exercise 4

Logical Stages-Testing - No logical stages or testing 0/10

Coding-Annotation - A bit of a confusing implementation and some odd syntax on the if and whiles, however, you've got some of the ideas 7/25

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: 54.5%

Test mark: 60%

OVERALL MARK FOR MODULE: 56%

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
