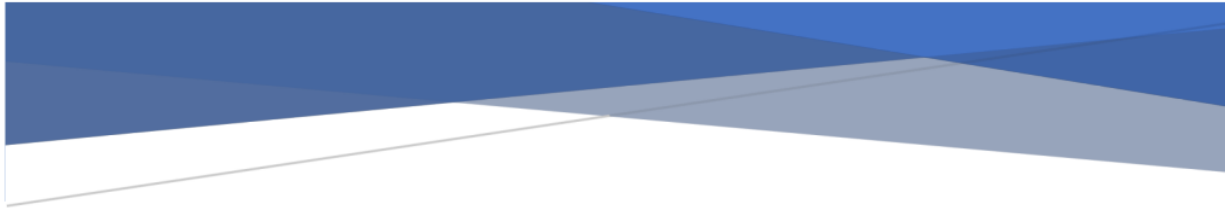


Computer Programming Assignment FINAL

by Josh Perry

FILE	19625_JOSH_PERRY_COMPUTER_PROGRAMMING_ASSIGNMENT_FINAL_34 1067_1084338013.DOCX (283.23K)		
TIME SUBMITTED	27-APR-2021 02:50PM (UTC+0100)	WORD COUNT	1448
SUBMISSION ID	150705960	CHARACTER COUNT	7430



FOUDNATION COMPUTER PROGRAMMING ASSIGNMENT 1

ACCA4020

Josh Perry (1903738)

Part A (Understanding)

Variables

A variable is a piece of information that is stored under a variable name. Figure 1 shows examples of variables and their various data types.

```
1 myNum = int(40) # Variable is of type integer
2 myPrice = float(3.99) # Variable is of type float
3 myName = str("Josh") # Variable is of type string
4 myList = [123, 456, 789] # Variable is of type list (lists integers)
```

Figure 1- Examples of Variables and their data types

Variables can be referenced and manipulated by a program for various purposes.

The sole purpose of a variable is to put a label on data and that data is stored in the memory to be used or changed throughout the program.

Variables are typically given appropriate names that reflect their purpose. E.g., a variable used to store my favourite fruit would have an appropriate name, following the correct conventions (camel casing [1]). Figure 2 shows an example of incorrect and correct variable names.

```
1 myVar = "apple" # Follows convention, but is not descriptive.
2 variable = "apple" # Does not follow convention and is not descriptive.
3 joshs favourite fruit to eat = "apple" # Does not follow convention, too long, but descriptive.
4 favFruit = "apple" # Follows convention, short and simple, descriptive. This is the ideal variable name.
```

Figure 2- Examples of correct and incorrect variable names

Python is colloquially known as a weakly typed language. Weakly typed infers that the syntax rules are more relaxed than a strongly typed language. This means the programmer has more freedom with type casting and converting between data types. In Perl, when declaring a variable, the programmer is expected to declare the datatype, however, in Python, if the programmer chooses not to define the data type, Python will attempt to distinguish the correct datatype.

While there is disagreement among experts on the definitions of weakly and strongly typed languages [2], the general consensus is that weakly typed languages allow the programmer to work around the language's imposed rules, whereas a strongly typed language offers less or no leeway in working around the rules.

If/Else Statements

To understand an if/else statement it is easier to break it down into two parts, the if and the else.

It is important to note that an If can run without an Else, but the reverse is not possible.

An If is used in programming to check if a condition is met. This is particularly useful if the programmer would like to validate information entered by the end user to avoid any crashes.

For example, figure 3 shows how an if statement is used alone.

```
myInt = int(input("please enter a number less than 30\n"))

if myInt <= 30:
    print("congratulations, your number is:", myInt)
```

Figure 3- example of an If Statement

As can be seen, the user is prompted to enter a number that is no greater than 30. The program will only print "Congratulations, your number is X" if the condition myInt is equal to or less than 30. However, if the condition is not met, then the program will exit, and the code inside the if statement will not be executed. This is an example of basic data validation, however, alone, this is a very weak validation system. To further improve the validation capabilities of this system, we need to add an Else. An Else allows the programmer to tell the program to take another action if the condition defined in the If statement is not met. Figure 4 shows the above code extended.

```
myInt = int(input("please enter a number less than 30\n"))

if myInt <= 30:
    print("congratulations, your number is:", myInt)
else:
    print("Sorry that number is too large, try again!")
```

Figure 4-example of an if/else statement

In this snippet, if the user enters a number larger than 30, the program will print "Sorry that number is too large, try again!". This allows for the program to determine which is the correct response expected from the user.

Loops

For Loop

A for loop is used when the programmer knows how many times the loop must run. This is because in order for a For loop to work, there needs to be a range. The range is the starting value, the end value and the step or increment. Appending the definition with the step is not always necessary, if a step is omitted, python will assume it is 1.

While Loop

In programming, while loops are incredibly useful. They work on the basis of executing the code contained within the loop, while a condition is true. This allows a program to loop for actions such as multiple guesses- ideal for entering of passwords or guessing games or in general actions that need to be repeated while a given condition is true. Figure 5 shows an example of a program, that asks the user to guess the magic number.

```

import random

myInt = 1
magicNo = int(random.randint(1, 10))
userIn = int(0)

userIn = int(input("guess the magic number!\n"))

while myInt != 3:
    if userIn != magicNo:
        myInt = myInt + 1
        userIn = int(input("Guess again!\n"))
    else:
        print("You guessed it!")
        exit()

if myInt == 3:
    print("Sorry, you lose this time!")

```

Figure 5- example program of a guessing game

In this example, the user is prompted to guess the magic number. The While loop determines how many attempts the user is allowed per game. In this instance, the user is granted 3 attempts to correctly guess the magic number. However, the magic number is a randomly generated number between 1 and 10 and is never shown to the user. Thus, the while loop allows the user multiple attempts. Within the loop, for every incorrect guess +1 is added to myInt, and eventually the myInt variable will be of value 3 and will cause the loop to exit and execute the next piece of code. This loop is also very useful as a counter function adding one to a value for every iteration of the loop.

Part B (Implementation)

Exercise 1

a. Logical Stages

1. Prompt user to enter a car make.
2. Add user input to variable carMake.
3. Prompt user to enter car model.
4. Add user input to carModel.
5. Prompt user to input car mileage.
6. Add user input to carMileage.
7. Prompt user to enter number of doors.
8. Add user input to carDoor.
9. Print all variables to the console.

b. Code

```
carMake = input("Please enter a car make:\n")
carModel = input("Please enter a car model:\n")
carMiles = input("Please enter the car's mileage:\n")
carDoors = input("Please enter the number of doors the
car has:\n")

print("Your", carMake, carModel, "Has", carMiles,
"Miles on the clock and", carDoors, "doors.")
```

c. Solution output

```
Please enter a car make:
ford
Please enter a car model:
focus
Please enter the car's mileage:
230000
Please enter the number of doors the car has:
5
Your ford focus Has 230000 Miles on the clock and 5 doors.
```

Exercise 2

a. Logical stages

1. Assign Value of 3.142 to varPi.
2. Prompt user to input the base measurement of the triangle and add to variable userBase.
3. Prompt user to input the base measurement of the triangle and add to variable userHeight.
4. Calculate area of triangle and store as finalTArea.
5. Print the final area of the triangle.
6. Prompt user to enter the radius of the circle and add to variable userRad.
7. Calculate the value of the userRad to the power of 2 and store as radSqr.
8. Calculate the final area of the circle and store as finalCArea.
9. Calculate the area of the circle minus the area of the triangle and store as endResult.
10. Print the final solution.

b. Code

```
varPi = float(3.142)
userBase = float(input("Please enter the base measurement
of triangle: "))
userHeight = float(input("Please enter the height
measurement of triangle: "))

finalTArea = (userBase * userHeight) / 2

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

userRad = float(input("Please enter the radius of the
circle:"))
radSqr = userRad ** 2
finalCArea = varPi * radSqr

endResult = finalCArea - finalTArea
print("Your final area is ", endResult)
```

c. Solution Output

```
Please enter the base measurement of triangle: 2
Please enter the height measurement of triangle: 5
The area of the triangle is: 5.0
Please enter the radius of the circle:10
Your final area is 309.2
```

Exercise 3

a. Logical Stages

1. Set up the products list.
2. Prompt input for list iterations.
3. Prompt input for each products price
4. Print a list of the products and prices entered.
5. Calculate the total price of all products.
6. Print to console.
7. Set the discount rate.
8. Calculate the discount amount.
9. Print the discount applied.
10. Calculate the discounted total.
11. Print the discounted total.

b. Code

```
products = [0, 0, 0, 0, 0]
products[0] = input("Please enter First product: \n")
products[1] = input("Please enter Second product: \n")
products[2] = input("Please enter Third product: \n")
products[3] = input("Please enter Fourth product: \n")
products[4] = input("Please enter Fifth product: \n")

firstPrice = float(input("Please enter the first price: \n"))
secondPrice = float(input("Please enter the second price: \n"))
thirdPrice = float(input("Please enter the third price: \n"))
fourthPrice = float(input("Please enter the fourth price: \n"))
fifthPrice = float(input("Please enter the fifth price: \n"))

print("Products Entered: \n")
print(products[0], firstPrice, "\n")
print(products[1], secondPrice, "\n")
print(products[2], thirdPrice, "\n")
print(products[3], fourthPrice, "\n")
print(products[4], fifthPrice, "\n")

pricesTotal = round(firstPrice + secondPrice + thirdPrice + fourthPrice + fifthPrice, 1)

print("Sub Total: ", pricesTotal)
discountRate = float(0.15)
discountAmount = pricesTotal * discountRate
print("Discount applied: ", round(discountAmount, 2))
discountedTotal = pricesTotal - discountAmount
print("your total is: ", round(discountedTotal, 2))
```

c. Solution Output

Please enter First product:

apple

Please enter Second product:

TV

Please enter Third product:

milk

Please enter Fourth product:

cheese

Please enter Fifth product:

tobacco

Please enter the first price:

0.89

Please enter the second price:

100.00

Please enter the third price:

1.40

Please enter the fourth price:

5

Please enter the fifth price:

23.99

Products Entered:

apple 0.89

TV 100.0

milk 1.4

cheese 5.0

tobacco 23.99

Sub Total: 131.3

Discount applied: 19.7

your total is: 111.61

References

- [1] P. Christensson, "CamelCase Definition," Sharpened Productions, 1 October 2020. [Online]. Available: <https://techterms.com/definition/camelcase>. [Accessed 2 March 2021].
- [2] Wikipedia, "Strong and weak typing," Wikipedia, 16 December 2020. [Online]. Available: https://en.wikipedia.org/wiki/Strong_and_weak_typing. [Accessed 10 March 2021].

Computer Programming Assignment FINAL

GRADEMARK REPORT

FINAL GRADE

50/100

GENERAL COMMENTS

Instructor

PART A

1. Variables - A good explanation with some good examples. 5/5

2.If/Else - The word you are looking for here is "decision" You also need to consider that there can be more than a binary choice, there can be 3 or more decisions made by including "Elif" in the statement. Good example shown. 4 /5

3. Loops - Good explanation of loops. Would be nice to have an example of a for loop as well as a while 4.5/5

PART B

Exercise 1

Logical Stages/Testing - Logical stages are OK, but you have not included validation. On the testing, you need to state whether the program behaved as intended - you have only tested the happy path here. What if the user entered a minus number for mileage? 3 /5

Coding/Annotation - You need to cast the variables for mileage and doors and perform validation on them. There is nothing to stop the user entering minus numbers for mileage or doors 6/10

Exercise 2

Logical Stages/Testing - Logical stages are OK, but again, there is no validation of input. This in turn makes the testing limited. The testing needs to be much more thorough. 6/10

Coding/Annotation - A working solution has been achieved, but only if the user uses "happy path" values. You need to validate the input (so that it is not below 0) and also consider the ratio of triangle to circle (so that the triangle is not outside the limits of the circle) 6/10

Exercise 3

Logical Stages/Testing - Same comments as above
6/10

Coding/Annotation - Strangely, you have not used a list for the prices. No loops are used which would make your program more efficient. You must use a loop with lists, otherwise there is no real point in using them at all, as there is no advantage over using individual variables. 9/15

Exercise 4 (Turtle)

Not attempted. 0/25

Note: It is considered bad practice to reference Wikipedia in academic reports, since it can be edited by anyone. Always find the original link (look at the references for the Wikipedia page to find this)

TOTAL: 49.5%

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9
