



University of Wales
Trinity Saint David

Title of report

SOFTWARE DEVELOPMENT (PYTHON)

Module Code: ACCB4017

RANGEL GEORGIEV STOITSOV

Student ID: 2029065

Lecturer: **SUNET FERNANDO**

Date submission:

29/04/2022

Table of Contents

INTRODUCTION:	3
PART A (UNDERSTANDING).....	3
1. What is a variable and why are they important in writing programs?.....	3
2. What an if/else statement is, how and why they are used?	4
3. Why are loops needed in a program and when you might use a “for” and a “while” loop?	6
PART B (IMPLEMENTATION).....	8
Exercise 1:.....	8
Exercise 2:	10
Exercise 3:	11
Exercise 3.1:	13
References:.....	15

INTRODUCTION:

PART A (UNDERSTANDING)

1. What is a variable and why are they important in writing programs?

In Python, a variable is a memory space set aside for storing values. In other words, a variable in a Python programme gives data to the computer for processing. When you add a value to a Python variable, it is created. Variables do not need to be defined for a certain type, and they can even alter types once they are set. [1]

These are:

- Integers - int
- Decimals – float
- Letters – string (str)

What is a variable in Python?

A variable in Python is a memory area reserved for storing values. In other words, a variable in a Python application sends data to the machine for processing. When you assign a value to a Python variable, it is created. Variables do not need to be defined for a certain type, and they can even modify types once they have been set.

How you create a variable in Python:

You use the "=" symbol to assign a variable. The name of the variable appears on the left, and the value to be placed in the variable appears on the right. The variable can be either a "string," a "integer," or a "float." A string is a character sequence. We also deal with text data in programming, such as names, messages, and explanations. Strings are used to describe such data in Python. String literals are written with single or double quotes ('hello there' or "hi there") on each side of the letter; keep in mind that the quotes are syntactic components, not part of the value. Single and double quotes are used to build a string. However, there are a few tiny variances.

Example 1:

name = "Rangel" – Example of a string.

A variable can be a "integer," which is the most fundamental sort of number. It can only be stated as a whole number, without the need of a decimal point. A "float" is a more sophisticated number type. A decimal number is known as a float.

Example 2: Integer and a float number:

age = 43 – “integer” example

age = 16.5 – “float” example

2. What an if/else statement is, how and why they are used?

When writing programmes, developers want to be able to make the best decisions based on the facts. They want the data to do what it's meant to do at the right time. In software development, this is referred to as conditional flow. When we use a conditional, how do we get data to act correctly? [2]

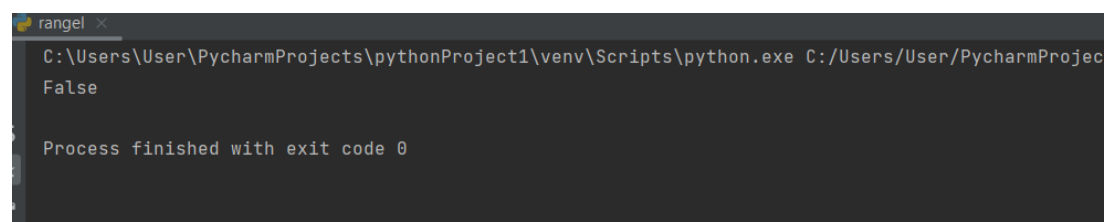
When the data approaches a conditional, the conditional instructs it on where to move based on certain current factors. Conditionals are created using "If" statements, comparison, and logical operators (, ">", "=", ">=", "==", "! =", "&&", "||"). Simple logical structures are described using the reserved phrases "if," "else," "elif," and "end." More sophisticated conditional statements will arise from more in-depth use of comparing operators. It's worth noting that all comparison operators provide a "boolean value." A “Boolean” meaning is either "true" or "false," with no other possibilities.

Examples of a comparison:

Example 1:

```
x = 6
y = 4
print (x == y)
```

returns False because 6 is not equal to 4



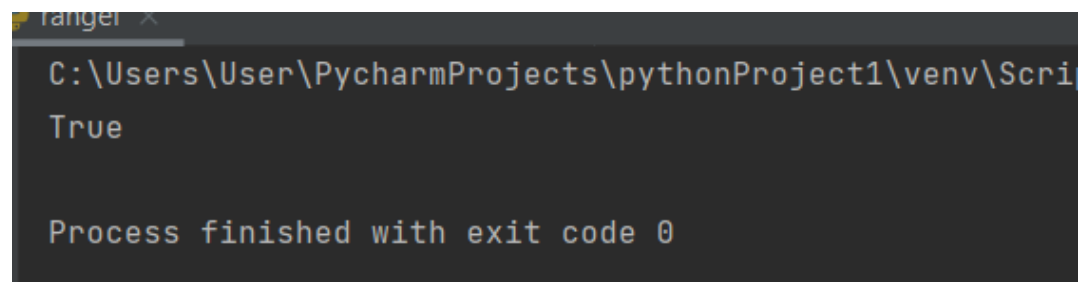
```
rangel x
C:\Users\User\PycharmProjects\pythonProject1\venv\Scripts\python.exe C:/Users/User/PycharmProjec
False

Process finished with exit code 0
```

Example 2:

```
x = 6
print(x > 3 and x < 10)
```

returns True because 6 is greater than 3 AND 6 is less than 10

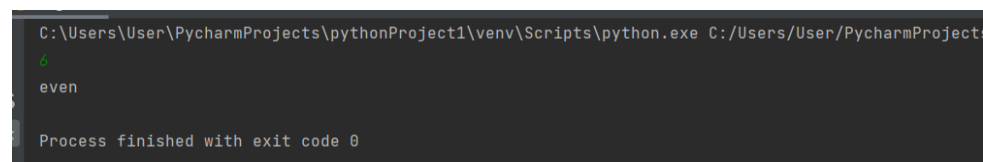


```
C:\Users\User\PycharmProjects\pythonProject1\venv\Scripts\python.exe
True
Process finished with exit code 0
```

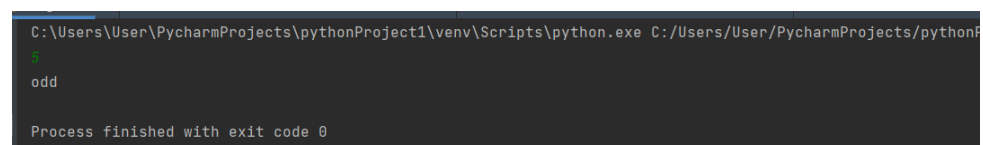
Example 3:

Write a program that checks whether an integer is even or odd.

```
number = int(input())
if number % 2 == 0:
    print("even")
else:
    print("odd")
```



```
C:\Users\User\PycharmProjects\pythonProject1\venv\Scripts\python.exe C:/Users/User/PycharmProjects/pythonProject1/venv/Scripts/python.exe
even
Process finished with exit code 0
```



```
C:\Users\User\PycharmProjects\pythonProject1\venv\Scripts\python.exe C:/Users/User/PycharmProjects/pythonProject1/venv/Scripts/python.exe
odd
Process finished with exit code 0
```

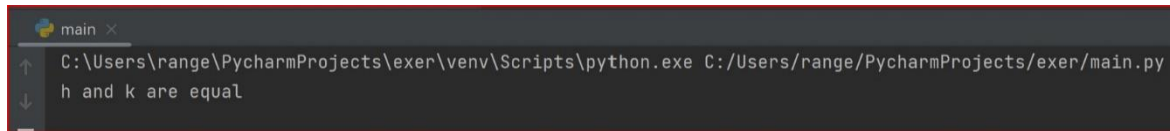
Example 4:

The “elif” keyword in Python means "try this condition if the previous conditions were not valid."

```

h = 31
k = 31
if k > h:
    print ("k is greater than h")
elif h == k:
    print ("h and k are equal")

```



```

main x
C:\Users\range\PycharmProjects\exer\venv\Scripts\python.exe C:/Users/range/PycharmProjects/exer/main.py
h and k are equal

```

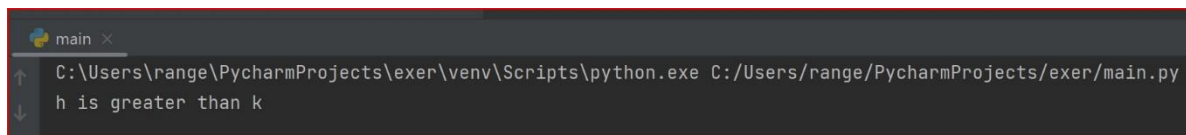
Example 5:

Anything that isn't caught by the previous criteria is caught by the “else” keyword.

```

h = 100
k = 22
if k > h:
    print ("k is greater than h")
elif h == k:
    print ("h and k are equal")
else:
    print ("h is greater than k")

```



```

main x
C:\Users\range\PycharmProjects\exer\venv\Scripts\python.exe C:/Users/range/PycharmProjects/exer/main.py
h is greater than k

```

3. Why are loops needed in a program and when you might use a “for” and a “while” loop?

The for loop allows a programmer to repeatedly repeat a collection of instructions; it also shortens the code, making loop variables easier to maintain. In Python, there are two types of loops: for and while. Both exist in Python, although the for loop in Python has a somewhat different focus. A while loop necessitates the developer repeating a single statement or a series of statements for the TRUE condition. Before beginning the loop, it verifies the condition.

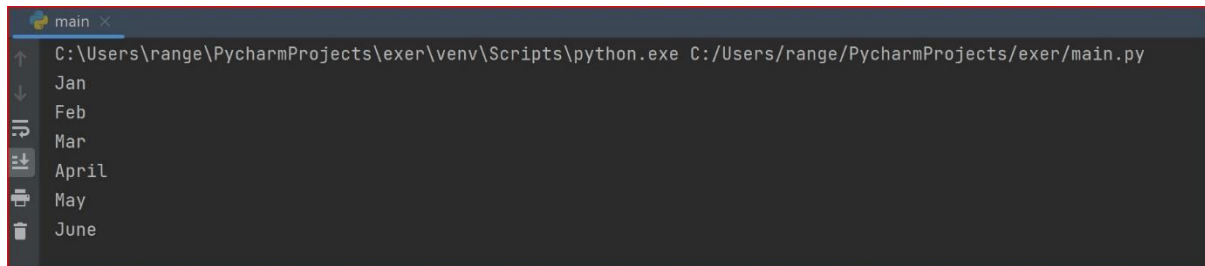
Examples of "for" and "while" loops include:

Example 1:

How to use For Loop for String

In this phase, we'll look at how "for loops" can be used for things other than numbers.

```
#use a for loop over a collection
Months = ["Jan","Feb","Mar","April","May","June"]
for m in Months:
    print(m)
```



I store the months (“Jan, Feb , Mar, April, May, June”) in variable Months, then I iterate the for loop over each value in Months. The current value of Months is stored in variable ‘m’. Print the month.

Example 2:

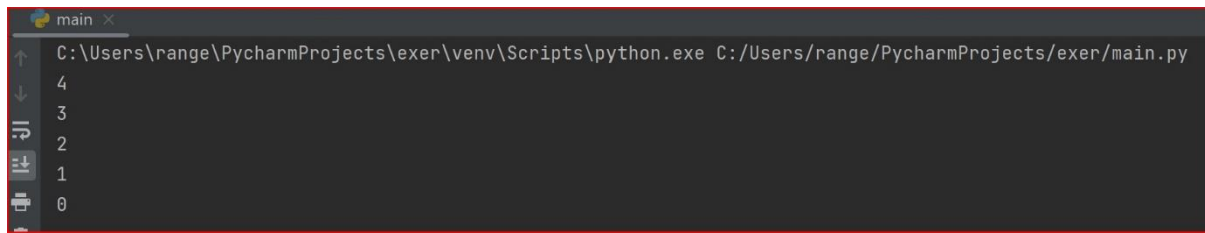
How to use while Loop

In this example, I'll show you what I mean: Initially, n equals 5. The expression in the while expression's header on line 2 is $n > 0$, which is true, hence the body of the loop is performed. Inside the body of row 3's cycle, n is decreased by one to four and then printed.

When the body of the cycle is finished, the programme returns to the beginning of the cycle of line 2 and the expression is evaluated once more. Because it is still true, the body is executed once more, and 3 is printed.

This process is repeated until n equals zero. When the expression is checked at this stage, it returns false, and the loop is terminated. Execution will restart at the first statement after the body of the cycle, but there is none in this case.

```
n = 5
while n > 0:
    n -= 1
    print(n)
```



```
main x
C:\Users\range\PycharmProjects\exer\venv\Scripts\python.exe C:/Users/range/PycharmProjects/exer/main.py
4
3
2
1
0
```

PART B (IMPLEMENTATION)

Exercise 1:

Demonstrate how variables can hold several types of values by defining three variables that will include a name, a height measurement (e.g. 1.84m), and a person's age (e.g. 26). Prompt the user to enter these values and display them on the screen. The variables should be suitably typed for the data that is required. The program's output could look like this.

1. Write a program to allow the user to input these values and then display them on the screen.

Logical stages

1. Prompt user to enter his name (expected to be a “string”)
2. Prompt user to enter his age (expected to be a “string”)
3. Prompt user to enter his height in metres (expected to be an integer)
4. Print result on the screen

Code with annotation

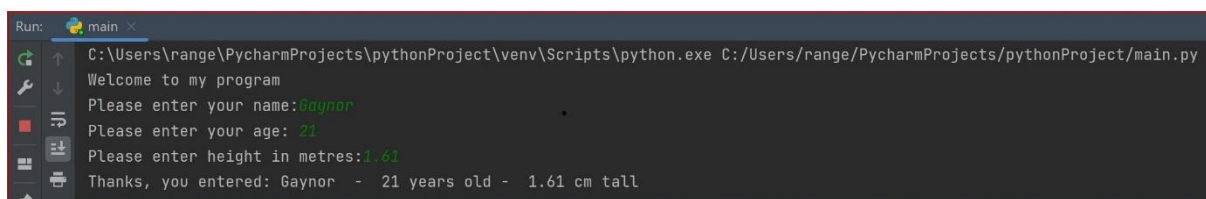
```
print("Welcome to my program")
name = str(input("Please enter your name:"))
age = int(input("Please enter your age: "))
height_in_metres = float(input("Please enter height in metres:"))
print("Thanks, you entered:" ,name, " - " ,age, "years old" " - " ,height_in_metres,
"cm tall")
```

Testing:

Normal values, within range:

```
Welcome to my program
Please enter your name: Gaynor
Please enter your age: 21
Please enter height in metres: 1.61
Thanks, you entered: Gaynor - 21 years old - 1.61 cm tall

Process finished with exit code 0
```



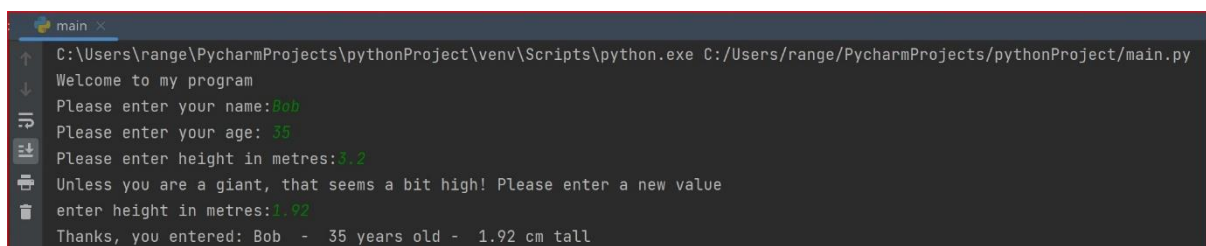
```
Run: main x
C:\Users\range\PycharmProjects\pythonProject\venv\Scripts\python.exe C:/Users/range/PycharmProjects/pythonProject/main.py
Welcome to my program
Please enter your name: Gaynor
Please enter your age: 21
Please enter height in metres: 1.61
Thanks, you entered: Gaynor - 21 years old - 1.61 cm tall
```

2. Create a software that will allow the user to enter these values and display them on the screen.

Result: Program performed as expected, correct answer calculated

Test the user input for out of range :

```
Welcome to my program
Please enter your name: Bob
Please enter your age: 35
Please enter height in metres: 3.2
Unless you are a giant, that seems a bit high! Please enter a new value
enter height in metres: 1.92
Thanks, you entered: Bob - 35 years old - 1.92 cm tall
```



```
main x
C:\Users\range\PycharmProjects\pythonProject\venv\Scripts\python.exe C:/Users/range/PycharmProjects/pythonProject/main.py
Welcome to my program
Please enter your name: Bob
Please enter your age: 35
Please enter height in metres: 3.2
Unless you are a giant, that seems a bit high! Please enter a new value
enter height in metres: 1.92
Thanks, you entered: Bob - 35 years old - 1.92 cm tall
```

Result: The user was not allowed to enter a value greater than 3 in place of height, so a message appears that he is a giant and must enter a valid value up to 3 meters high. The program is executed correctly.

Exercise 2:

Allow the user to enter the measurements (in cm) of the forms illustrated below, then calculate the area of the blue shaded region. It should be noted that the square and rectangular regions are both empty.

Logical stages

1. Invite the user to enter the radius of the circle (expected to be an "integer")
2. Prompt the user to enter the length of the side of the square (expected to be an "integer")
3. Invite the user to enter the width of the rectangle (expected to be an "integer")
4. Invite the user to enter the side of the rectangle (expected to be an "integer")
5. Calculate the available area of the circle by removing the area of the square and the circle.
6. Print the result on the screen

Code with annotation

```
from math import pi

r = float(input("Input the radius of the circle: "))
r = pi * r ** 2
b = float(input("Please enter side of square: "))
c = float(input("Please enter width of rectangle: "))
d = float(input("Please enter length of rectangle: "))
square=b*b
rectangle=c*d
shaded_area = r - (square+rectangle)

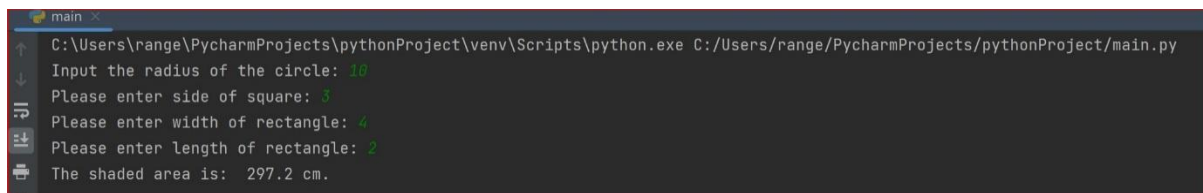
print("The shaded area is: ", round(shaded_area, 1), str("cm."))
```

Testing the program for the good scenario.

Result: Program performed as expected, correct answer calculated the shaded area.

Input the radius of the circle: 10
Please enter side of square: 3
Please enter width of rectangle: 4
Please enter length of rectangle: 2
The shaded area is: 297.2 cm.

Process finished with exit code 0



```
main x
C:\Users\range\PycharmProjects\pythonProject\venv\Scripts\python.exe C:/Users/range/PycharmProjects/pythonProject/main.py
Input the radius of the circle: 10
Please enter side of square: 3
Please enter width of rectangle: 4
Please enter length of rectangle: 2
The shaded area is: 297.2 cm.
```

Exercise 3:

Write a program that allows the user to car mileage (per day) for a week. When the mileage has been entered, calculate the total mileage. Finally, calculate the total cost of petrol for the week, at the rate of £1.35/litre and 11.2 miles to the litre. Use a list/array and loop(s) where appropriate.

Logical stages

1. Invite the user to enter the miles travelled Monday (may not be an integer)
 2. Invite user to enter mileage Tuesday (may not be an integer)
 3. Invite the user to enter the miles travelled Wednesday (may not be an integer)
 4. Invite the user to enter the miles travelled Thursday (may not be an integer)
 5. Invite user to enter mileage Friday (may not be an integer)
 6. Invite the user to enter the miles travelled Saturday (may not be an integer)
 7. Invite user to enter mileage Sunday (may not be an integer)
 8. Calculate the miles travelled for the week and calculate the amount of fuel for the miles travelled.
6. Print the result on the screen.

Code with annotation:

```
def is_digit(string): # function to check if str is float or number
    if string.isdigit():
        return True
    else:
        try:
            float(string)
            return True
        except ValueError:
            return False

def insertMiles(): # Function to read miles from the keyboard
    miles = input() # We read the number on the keyboard

    if is_digit(miles): # Let's check if it's a number
        miles = float(miles) # We transform the string into int
    else: # If there is no number:
        print("Please try again") # I was displaying the message
        miles = insertMiles() # We're reading another number on the keyboard

    if miles < 0: # Check if the number is not negative
        print("Please try again...") # I'm posting the message
        miles = insertMiles() # We're reading the number again

    return miles #

if "main" == "main":
    rate = 1.35 # We declare the values needed to calculate consumption
    consumption = 11.2 # We declare the values needed to calculate
consumption
    WeekDays = ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday",
"Saturday",
                "Sunday"] # we declare the days of the week

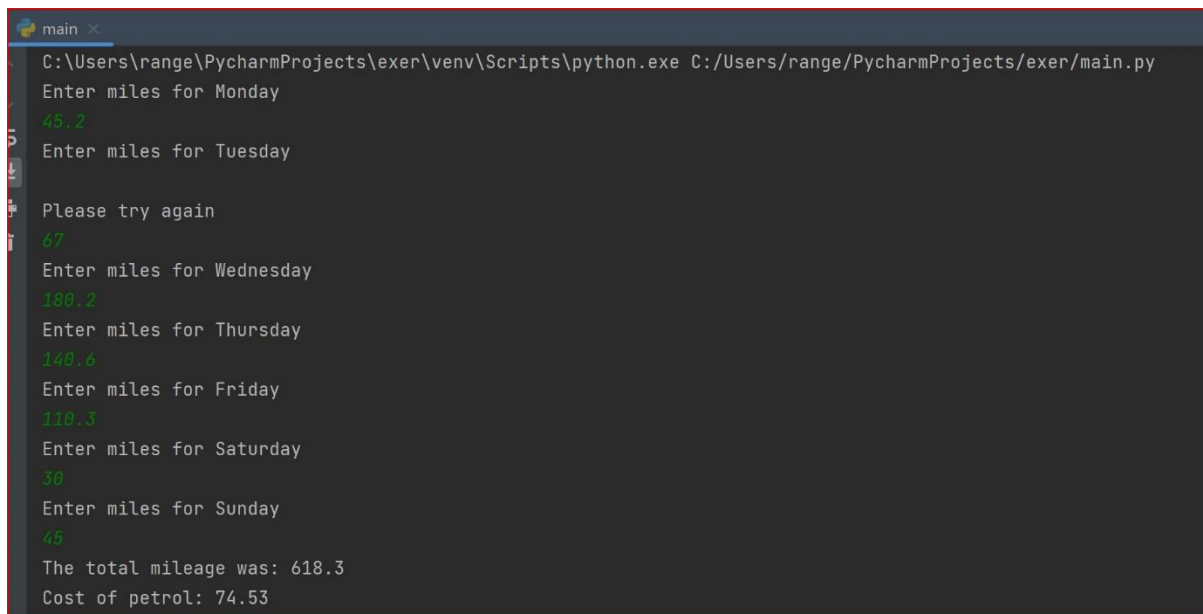
    totalMileage = 0 # declare our count
    for day in WeekDays: # we go through the massif with the days of the
week
        print("Enter miles for {0}".format(day)) # I was displaying the
message
        totalMileage += insertMiles() # We collect what we went for today

    rez = (totalMileage / consumption) * rate # We calculate the total
expenses
    # I was displaying the message
    print("The total mileage was: {0}".format(round(totalMileage, 1)))
    print("Cost of petrol: {0}".format(round(rez, 2))) # I was displaying
the message
```

Testing the program for the good scenario.

```
Enter miles for Monday :  
45.2  
Enter miles for Tuesday :  
67  
Enter miles for Wednesday :  
180.2  
Enter miles for Thursday :  
140.6  
Enter miles for Friday :  
110.3  
Enter miles for Saturday :  
30  
Enter miles for Sunday :  
45  
The total mileage was: 618.3  
Cost of petrol: 74.53
```

Result: The program runs as expected, the correct answer is calculated mileage and fuel cost



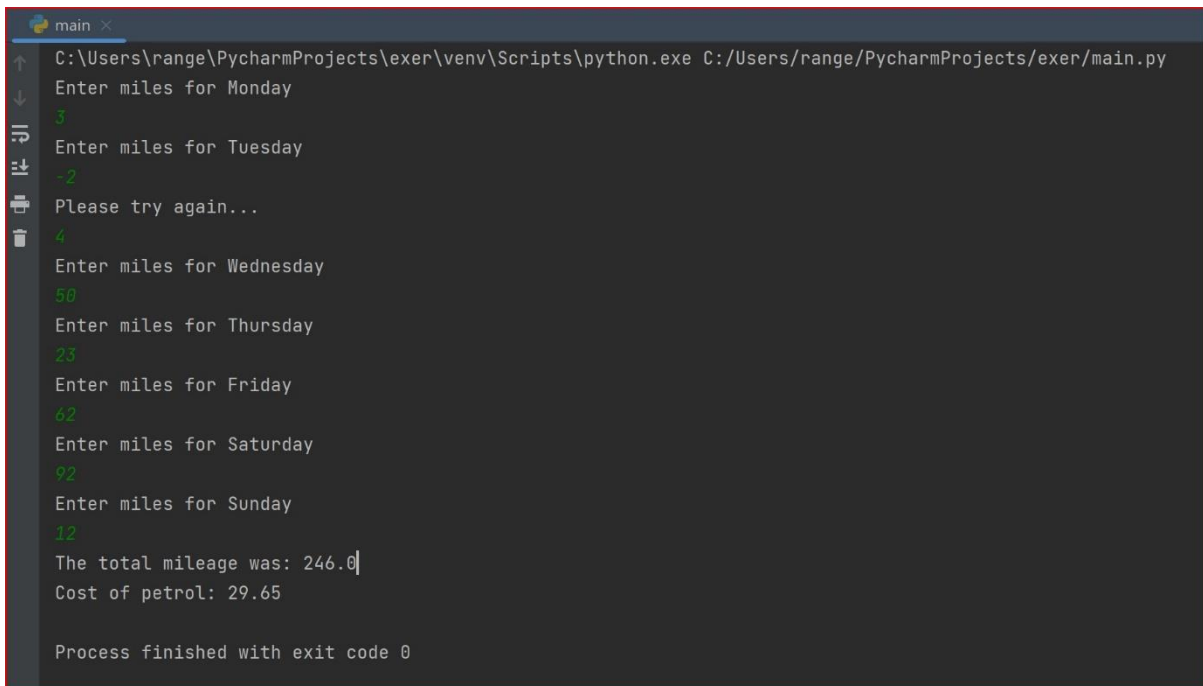
```
main x  
C:\Users\rangle\PycharmProjects\exer\venv\Scripts\python.exe C:/Users/range/PycharmProjects/exer/main.py  
Enter miles for Monday  
45.2  
Enter miles for Tuesday  
  
Please try again  
67  
Enter miles for Wednesday  
180.2  
Enter miles for Thursday  
140.6  
Enter miles for Friday  
110.3  
Enter miles for Saturday  
30  
Enter miles for Sunday  
45  
The total mileage was: 618.3  
Cost of petrol: 74.53
```

Exercise 3.1:

Testing the program for the good scenario.

```
Enter miles for Monday :  
3  
Enter miles for Tuesday :  
-2  
Please try again  
Enter miles for Tuesday :  
4  
Enter miles for Wednesday :  
50  
Enter miles for Thursday :  
23  
Enter miles for Friday :  
62  
Enter miles for Saturday :  
92  
Enter miles for Sunday :  
12  
The total mileage was: 246.0  
Cost of petrol: 29.65
```

Result: The program runs as expected, the correct answer is calculated mileage and fuel cost.



```
main x  
C:\Users\rangle\PycharmProjects\exer\venv\Scripts\python.exe C:/Users/rangle/PycharmProjects/exer/main.py  
Enter miles for Monday  
3  
Enter miles for Tuesday  
-2  
Please try again...  
4  
Enter miles for Wednesday  
50  
Enter miles for Thursday  
23  
Enter miles for Friday  
62  
Enter miles for Saturday  
92  
Enter miles for Sunday  
12  
The total mileage was: 246.0  
Cost of petrol: 29.65  
  
Process finished with exit code 0
```

References:

- [1] <https://github.com/SoftUni/Programming-Basics-Book-Python-BG>
- [2] [Python Tutorial \(w3schools.com\)](https://www.w3schools.com/python/)