Python – condition and control statements

Chapter 3 - Textbook

Topics include

- Control statements
- Control loops with fixed range of values
- Control loops with start and end boundaries
- Control loops with start and end boundaries and defined step size
- Control loop that will have start and end boundaries and step size is defined with negative (reverse)
- Control loop with conditions defined
- Control loops with defined condition to break the loop
- Conditional statements (multiple conditions)

Strings and sub-strings – also posted in lecture1

- Sequence of characters saved in contiguous memory locations ending with end-of-string character
- ‘This is string’ – 14 character including two spaces
- str1 = “This is string” – len(str1) will result in an integer value 14
- filename=“myfile.txt” – print (filename) will print myfile.txt
- filename=“myfile.txt” – print (filename[2:5]) prints fil (note index 5 is NOT printed, it prints from character at index 2 (‘f’) to character at index 4 (‘l’)
- Print (filename[-3:-2]) – here character at index -3 is ‘t’ (third character from the end) and character at index -2 is ‘x’

Sub-strings

```
fileList = ["myfile.txt", "myprogram.exe", "anotherfile.txt"]

for filename in fileList:
   if ".exe" in filename:
      print (filename)
```

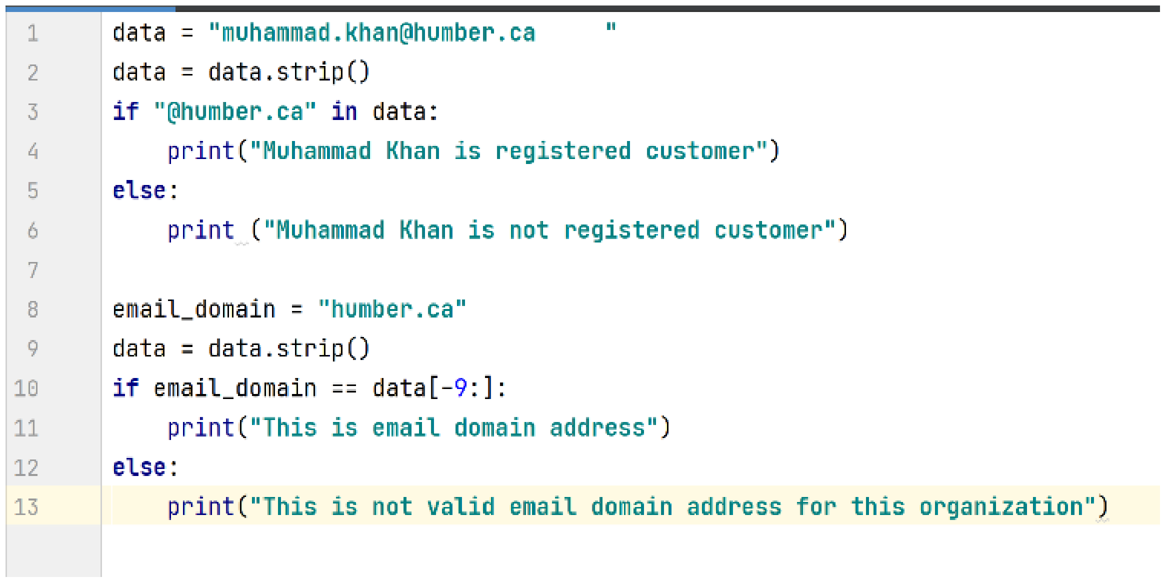
```
name= "muhammad khan"
print(name.upper())
city='TORONTO'
print(city.lower())
print(name.capitalize())
print(len(name)) –prints number
                   of characters
```

```
MUHAMMAD KHAN
toronto
Muhammad khan

Process finished with exit code 0
```

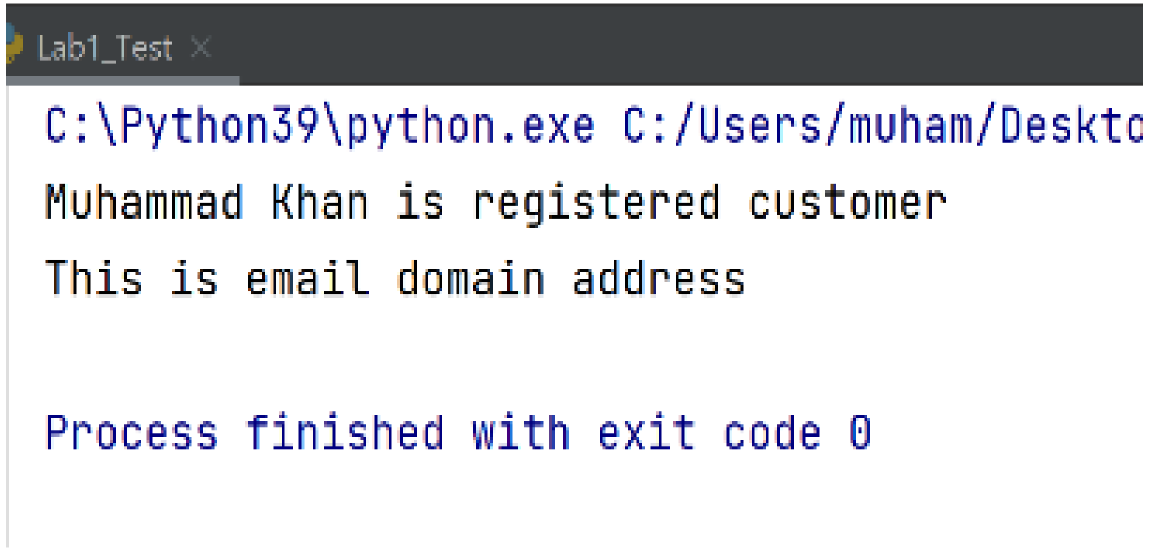
String functions

- split (‘ ‘) – creates a list of strings (after taking each string that is separated by a space.
- split(‘,’) – creates a list of strings, after splitting strings that was separated by ‘,’
- strip() – removes white spaces from the ends

Application of String functions



```python
data = "muhammad.khan@humber.ca     "
data = data.strip()
if "@humber.ca" in data:
    print("Muhammad Khan is registered customer")
else:
    print ("Muhammad Khan is not registered customer")

email_domain = "humber.ca"
data = data.strip()
if email_domain == data[-9:]:
    print("This is email domain address")
else:
    print("This is not valid email domain address for this organization")
```



```
C:\Python39\python.exe C:/Users/muham/Deskto
Muhammad Khan is registered customer
This is email domain address

Process finished with exit code 0
```

Loop – fixed number of times

Value of range varies from 0 to 3

```
for eachPass in range (4):
    print("Python is fun language...\t\t")
```

```
Python is fun language...
Python is fun language...
Python is fun language...
Python is fun language...
```

Loop - count

```
for eachPass in range (4):
    print(str(eachPass) + "\t Python is fun language...\t\t")
```

```
0		Python is fun language...
1		Python is fun language...
2		Python is fun language...
3		Python is fun language...
```

Sum of range of numbers

```
lower_bound = int (input ('Enter lower bound:  '))
upper_bound = int (input ('Enter lower bound: '))
sum = 0
for values in range (lower_bound, (upper_bound+1)):
    sum = sum + values
print ( sum )
```

```
Enter lower bound:  4
Enter lower bound: 8
30
```

Range of numbers – perception?

```
## Printing the range of values

for value in range (2,5):
    print(value)
```

```
2
3
4
```

Printing list of numbers

Use of ‘square brackets’

```
for number in [2,8,4,9,6]:
    print number
```

```
2
8
4
9
6
```

Printing range of numbers with step size

```
for values in range (2,10,2):
    print(values)
```

```
2
4
6
8
```

Printing in reverse

```
for values in range (10,2,-2):
    print(values)
```

```
10
8
6
4
```

Printing in Reverse and sum

```
sum=0
for values in range (10,2,-2):
    print(values)
    sum=sum+values
print ("sum =  " + str(sum) )
```

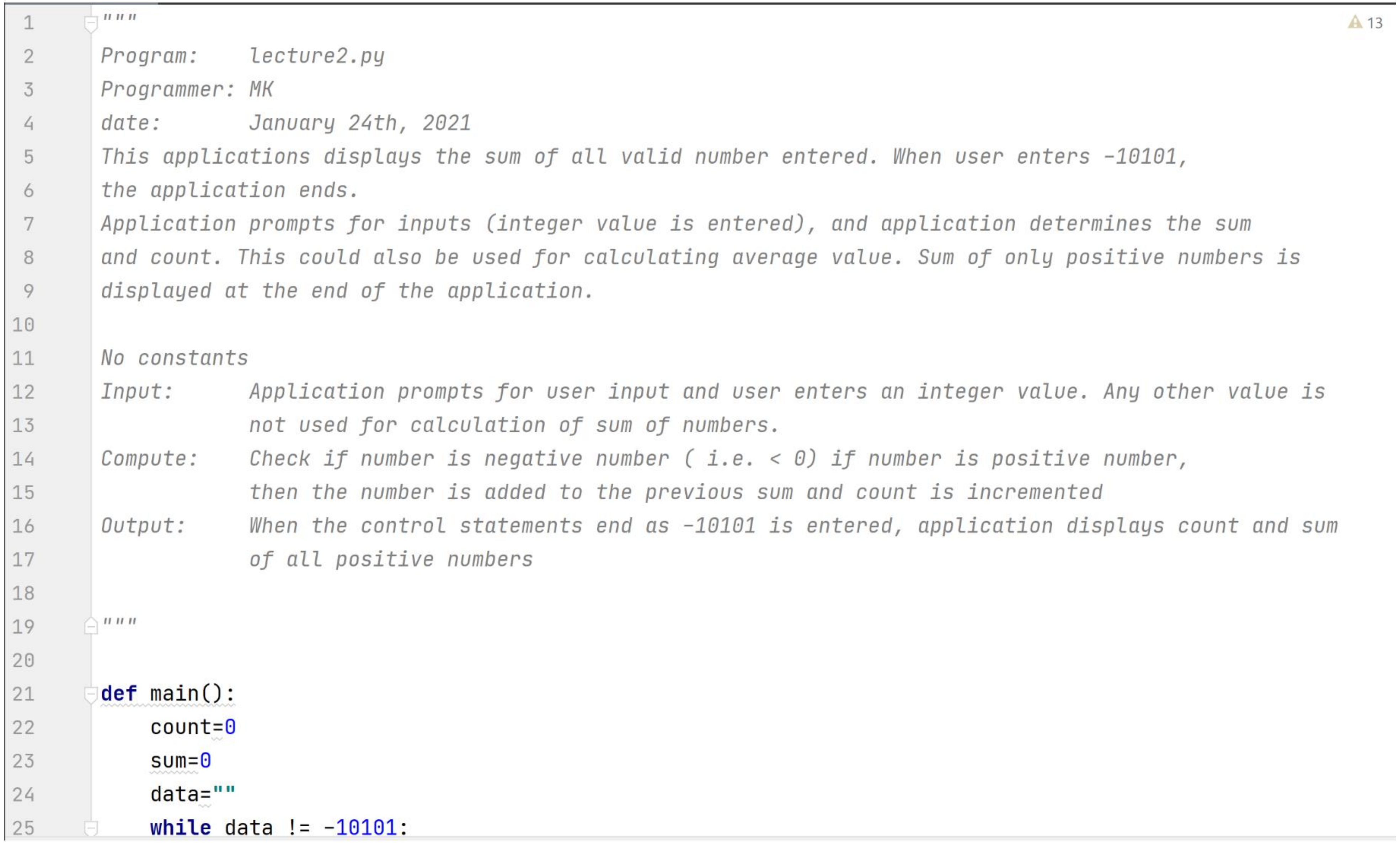
```
10
8
6
4
sum =  28
```

Control Statement – while loop

```
while condition:
        <statement(s)>
```

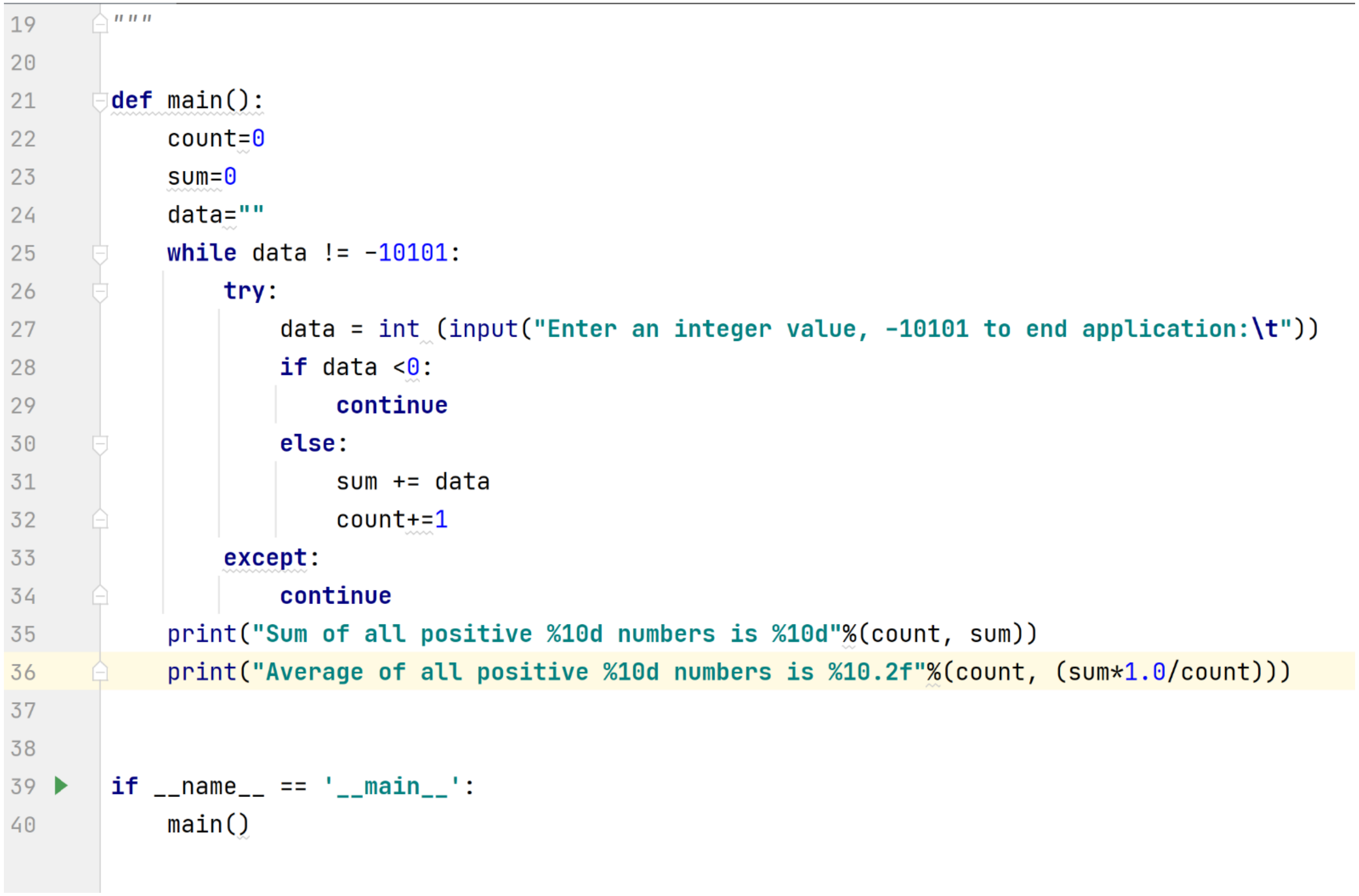
- Use of continue statement
- Use of break statement

Application that displays sum of all positive integer numbers and if user enters -10101, application ends



```
"""
Program:    lecture2.py
Programmer: MK
date:       January 24th, 2021
This applications displays the sum of all valid number entered. When user enters -10101,
the application ends.
Application prompts for inputs (integer value is entered), and application determines the sum
and count. This could also be used for calculating average value. Sum of only positive numbers is
displayed at the end of the application.

No constants
Input:      Application prompts for user input and user enters an integer value. Any other value is
            not used for calculation of sum of numbers.
Compute:    Check if number is negative number ( i.e. < 0) if number is positive number,
            then the number is added to the previous sum and count is incremented
Output:     When the control statements end as -10101 is entered, application displays count and sum
            of all positive numbers

"""

def main():
    count=0
    sum=0
    data=""
    while data != -10101:
```

Source Code for the problem stated on last slide



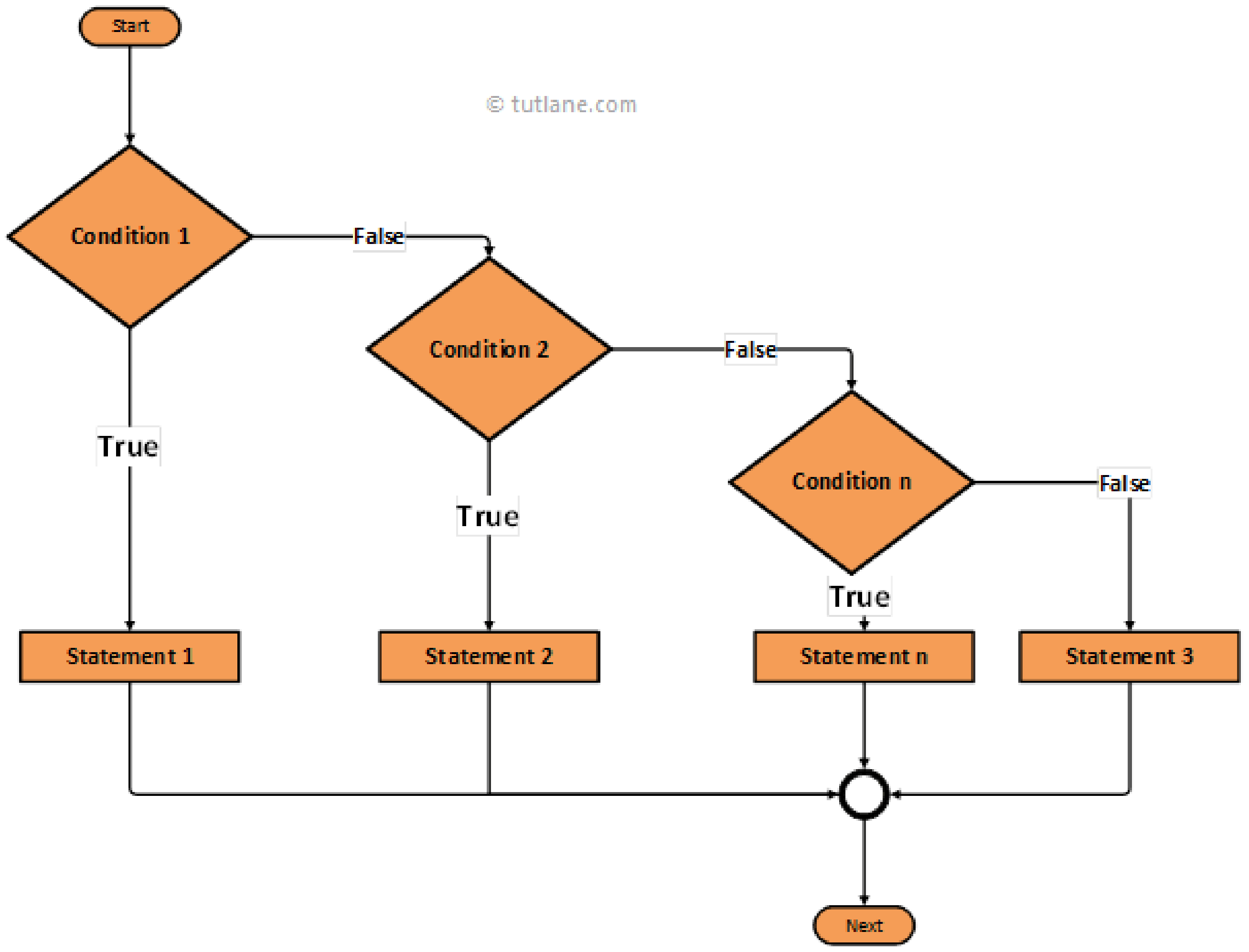
```python
"""

def main():
    count=0
    sum=0
    data=""
    while data != -10101:
        try:
            data = int (input("Enter an integer value, -10101 to end application:\t"))
            if data <0:
                continue
            else:
                sum += data
                count+=1
        except:
            continue
    print("Sum of all positive %10d numbers is %10d"%(count, sum))
    print("Average of all positive %10d numbers is %10.2f"%(count, (sum*1.0/count)))


if __name__ == '__main__':
    main()
```

Infinite Loop (using while)

```
while   True:
    <statements>
```

You may break loop under condition or you may continue the loop under a different condition

Graphical representation or Flow control statements based on conditions



Simple Boolean Expressions

==	Equals
!=	Not Equals
<	Less than
>	Greater than
<=	Less than OR equal to
>=	Greater than OR equal to

Maximum and smaller value:

```
first_number = int (input ('Enter first number:   '))
second_number = int (input('Enter second number:     '))

if  first_number > second_number:
    maximum= first_number
    smaller = second_number
else:
    maximum = second_number
    smaller = first_number

print("Maximum Value is:    " + str (maximum))
print("Smaller of the two values is:    "  + str (smaller))
```

```
Enter first number:   -45
Enter second number:     -10
Maximum Value is:   -10
Smaller of the two values is:   -45
```

Conditional statements – if – elif - else

```
mark = int (input('Enter mark:     '))
if mark >= 90:
    grade='A'
elif mark >=80:
    grade = 'B'
elif mark >= 70:
    grade = 'C'
elif mark >= 60:
    grade = 'D'
else:
    grade = 'F'
print ("Marks entered  " + str (mark) + " and calculated grade is " + str(grade))
```

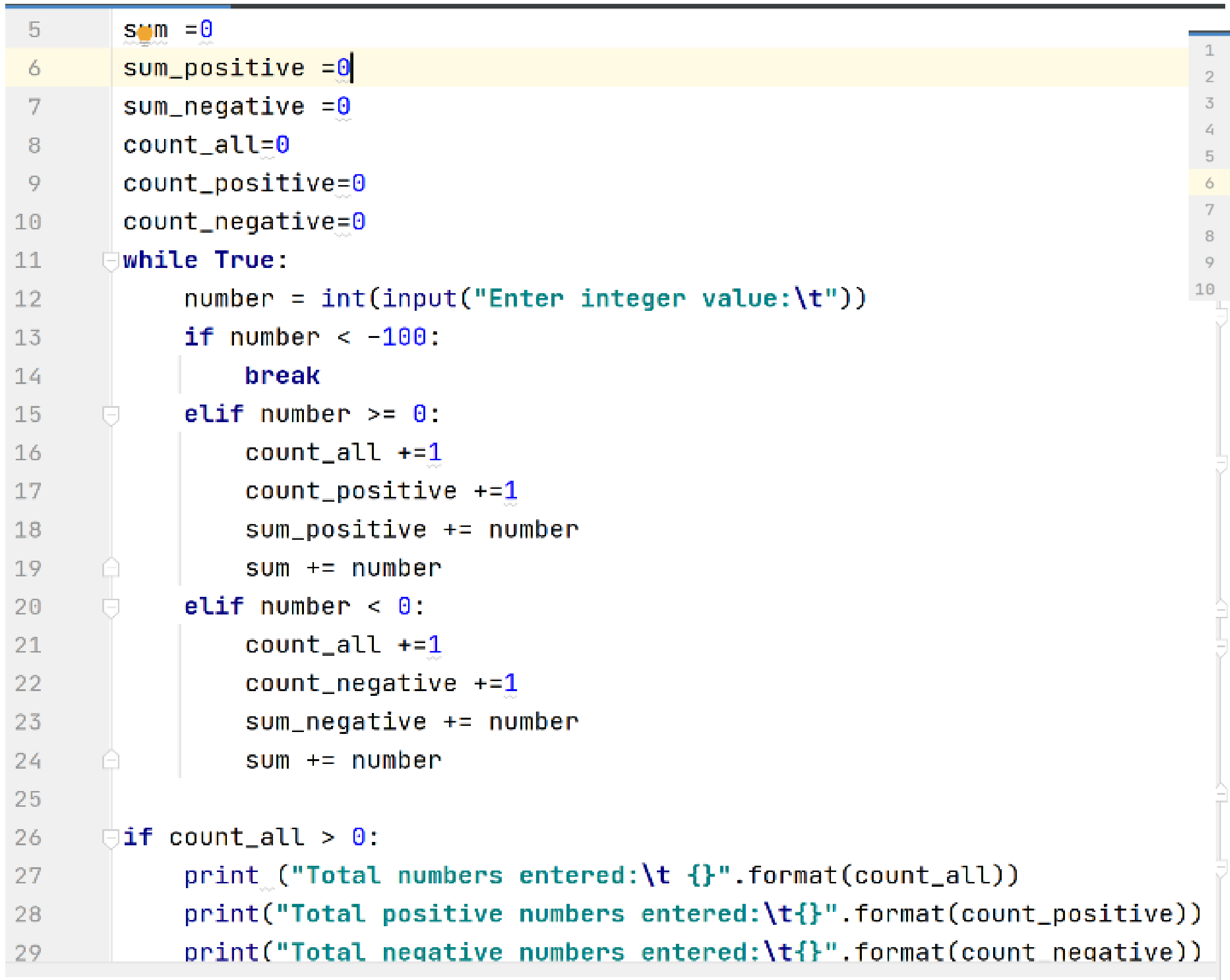
Multiple conditions

```
mark_entered = int (input ('Enter mark that must be between 0 and 100, both values inclusive....'))
if mark_entered < 0 or mark_entered > 100:
    print("You entered the score " + str(mark_entered) + " which not correct - mark must be between 0 and 100 ")
```

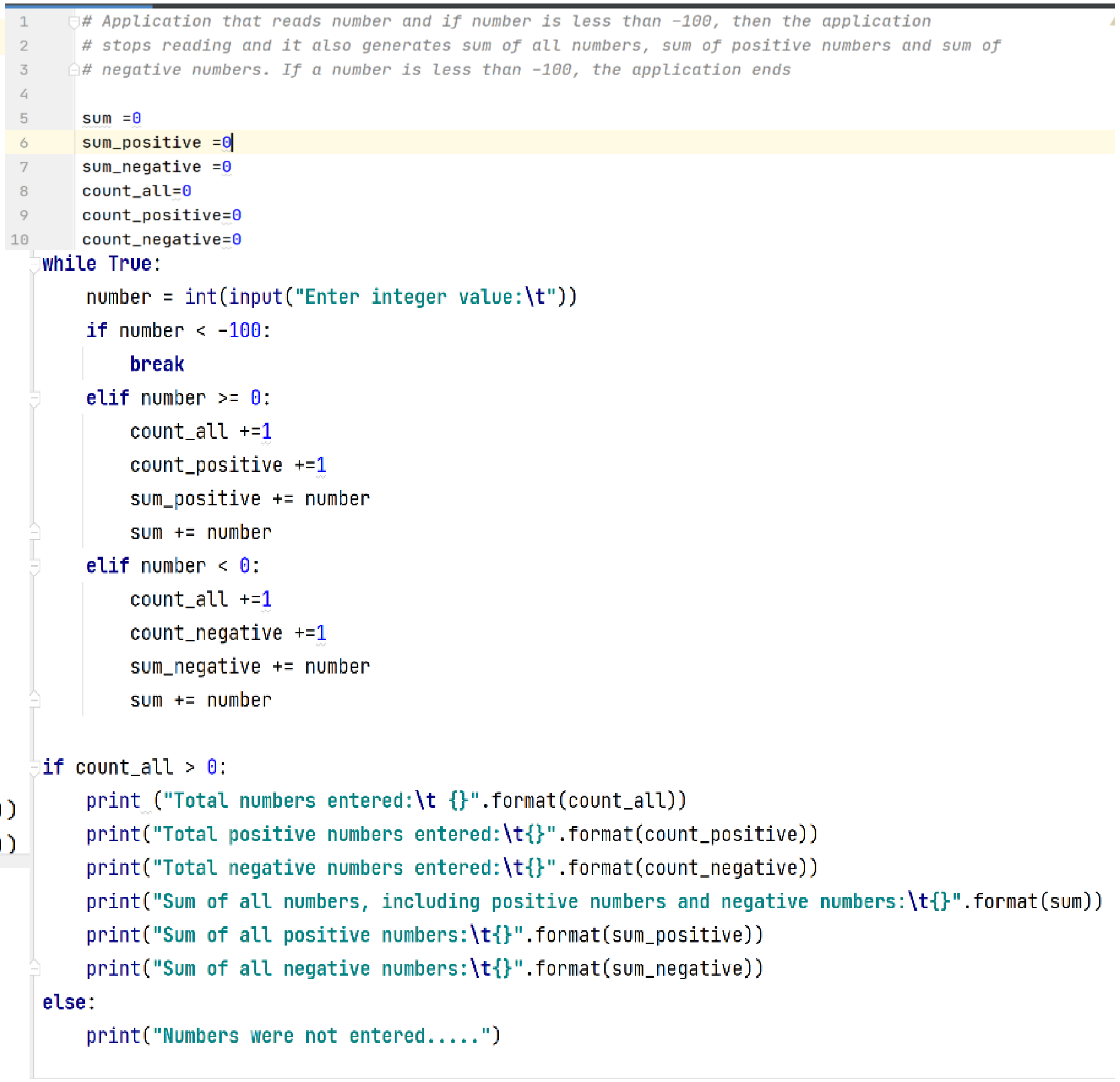
```
Enter mark that must be between 0 and 100, both values
inclusive....101
You entered the score 101 which must be between 0 and 100

Process finished with exit code 0
```

```
mark_entered = int (input ('Enter mark that must be between 0 and 100, both values inclusive....'))
if mark_entered >= 0 and mark_entered <= 100:
    print("You entered the score " + str(mark_entered) + " which valid score - mark is between 0 and 100 ")
else:
    print('Valid mark is between 0 and 100 - both 0 and 100 inclusive..........')
```

Conditional and Control statements

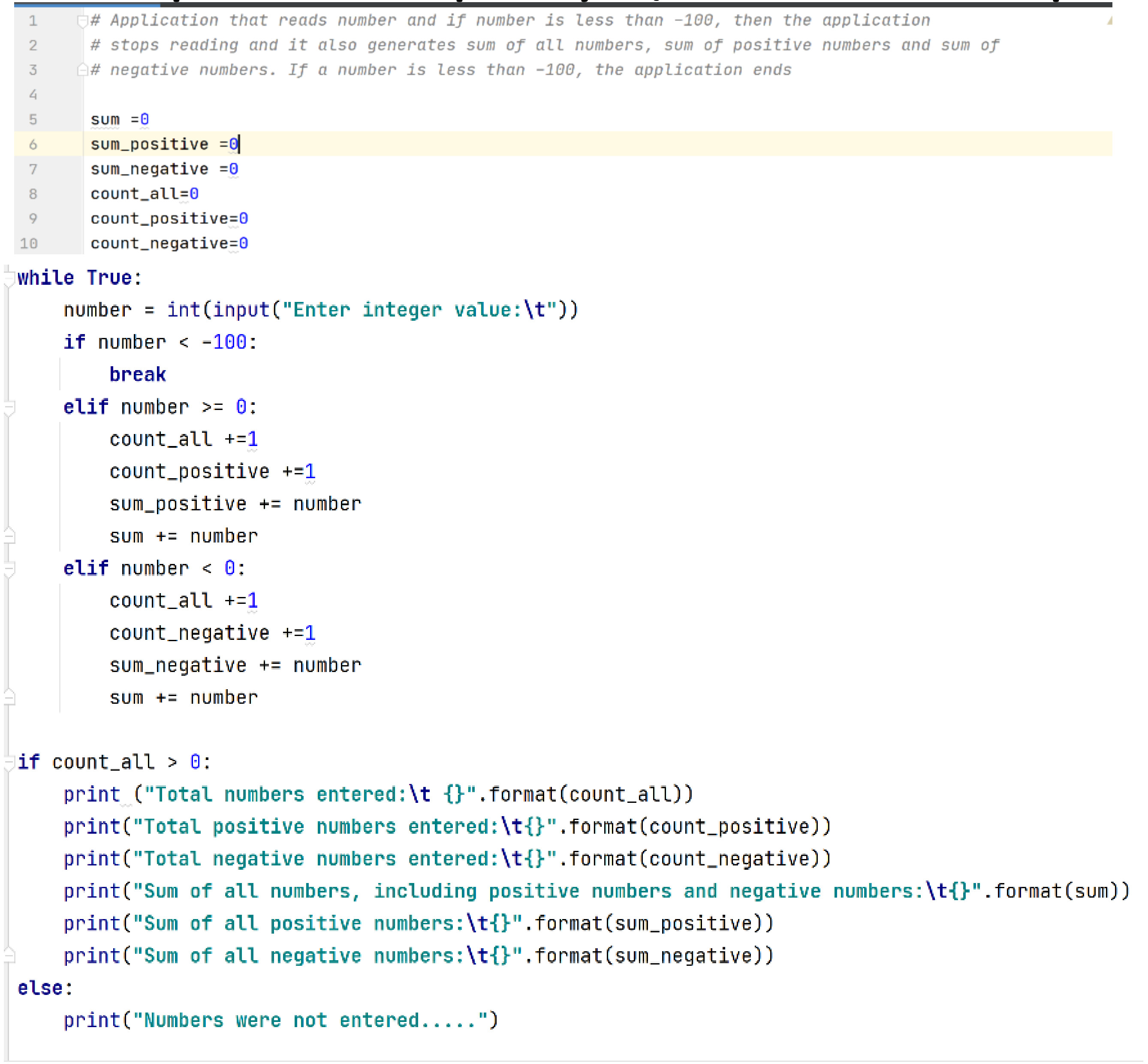


```python
sum =0
sum_positive =0
sum_negative =0
count_all=0
count_positive=0
count_negative=0
while True:
    number = int(input("Enter integer value:\t"))
    if number < -100:
        break
    elif number >= 0:
        count_all +=1
        count_positive +=1
        sum_positive += number
        sum += number
    elif number < 0:
        count_all +=1
        count_negative +=1
        sum_negative += number
        sum += number

if count_all > 0:
    print ("Total numbers entered:\t {}".format(count_all))
    print("Total positive numbers entered:\t{}".format(count_positive))
    print("Total negative numbers entered:\t{}".format(count_negative))
```

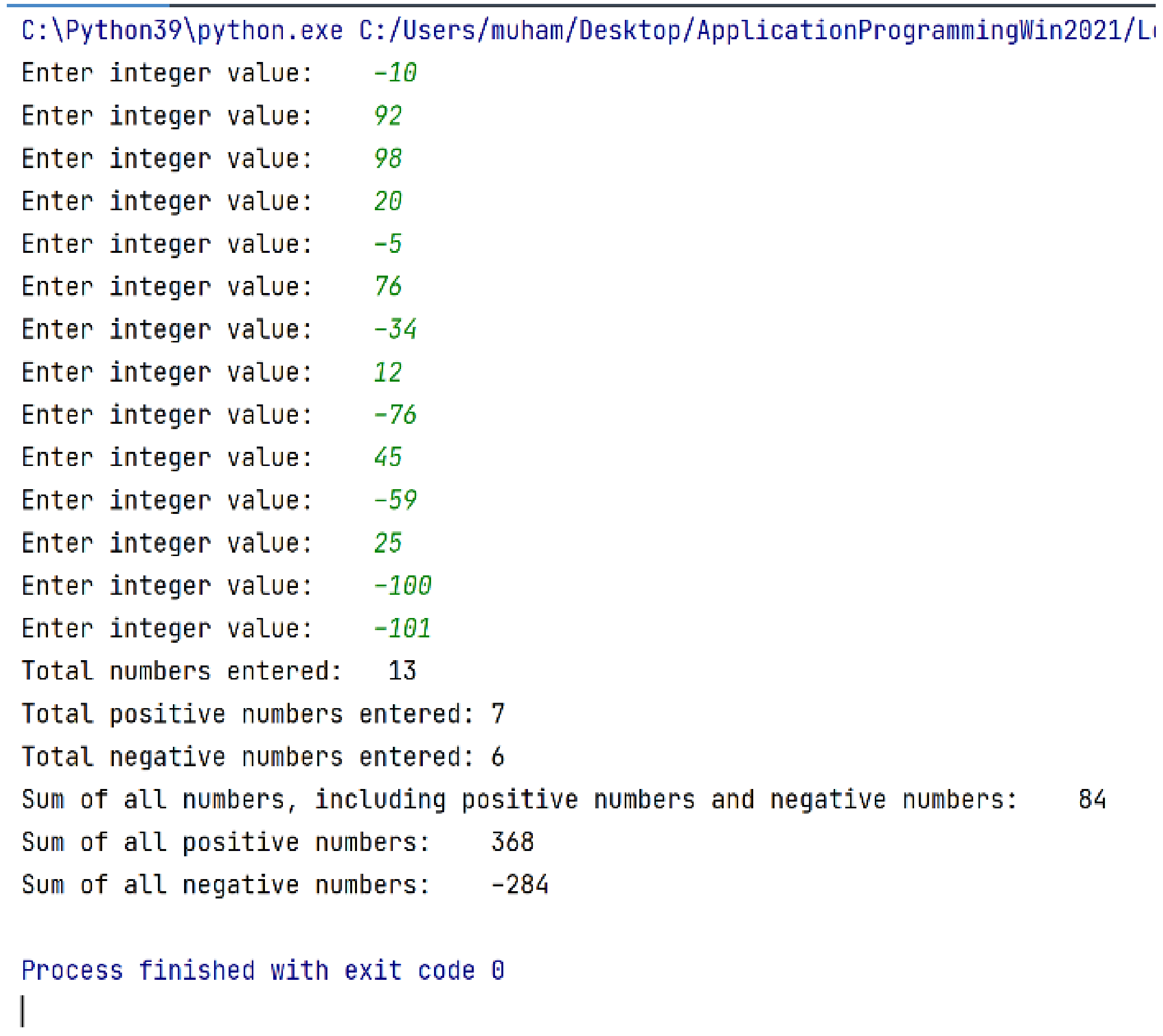


```python
# Application that reads number and if number is less than -100, then the application
# stops reading and it also generates sum of all numbers, sum of positive numbers and sum of
# negative numbers. If a number is less than -100, the application ends

sum =0
sum_positive =0
sum_negative =0
count_all=0
count_positive=0
count_negative=0
while True:
    number = int(input("Enter integer value:\t"))
    if number < -100:
        break
    elif number >= 0:
        count_all +=1
        count_positive +=1
        sum_positive += number
        sum += number
    elif number < 0:
        count_all +=1
        count_negative +=1
        sum_negative += number
        sum += number

if count_all > 0:
    print ("Total numbers entered:\t {}".format(count_all))
    print("Total positive numbers entered:\t{}".format(count_positive))
    print("Total negative numbers entered:\t{}".format(count_negative))
    print("Sum of all numbers, including positive numbers and negative numbers:\t{}".format(sum))
    print("Sum of all positive numbers:\t{}".format(sum_positive))
    print("Sum of all negative numbers:\t{}".format(sum_negative))
else:
    print("Numbers were not entered.....")
```

Output Display (last example)

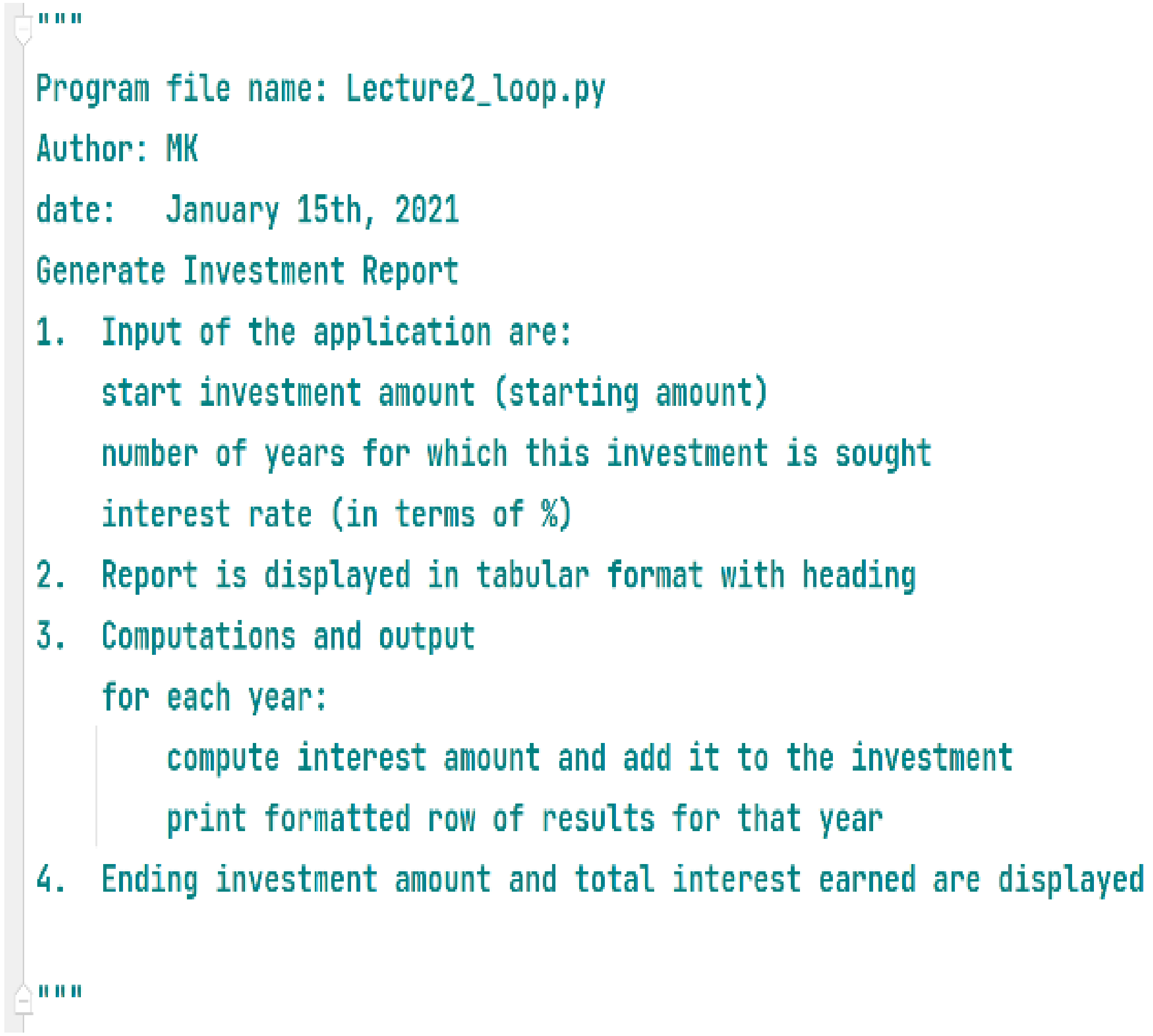


```python
# Application that reads number and if number is less than -100, then the application
# stops reading and it also generates sum of all numbers, sum of positive numbers and sum of
# negative numbers. If a number is less than -100, the application ends

sum =0
sum_positive =0
sum_negative =0
count_all=0
count_positive=0
count_negative=0
while True:
    number = int(input("Enter integer value:\t"))
    if number < -100:
        break
    elif number >= 0:
        count_all +=1
        count_positive +=1
        sum_positive += number
        sum += number
    elif number < 0:
        count_all +=1
        count_negative +=1
        sum_negative += number
        sum += number

if count_all > 0:
    print ("Total numbers entered:\t {}".format(count_all))
    print("Total positive numbers entered:\t{}".format(count_positive))
    print("Total negative numbers entered:\t{}".format(count_negative))
    print("Sum of all numbers, including positive numbers and negative numbers:\t{}".format(sum))
    print("Sum of all positive numbers:\t{}".format(sum_positive))
    print("Sum of all negative numbers:\t{}".format(sum_negative))
else:
    print("Numbers were not entered.....")
```



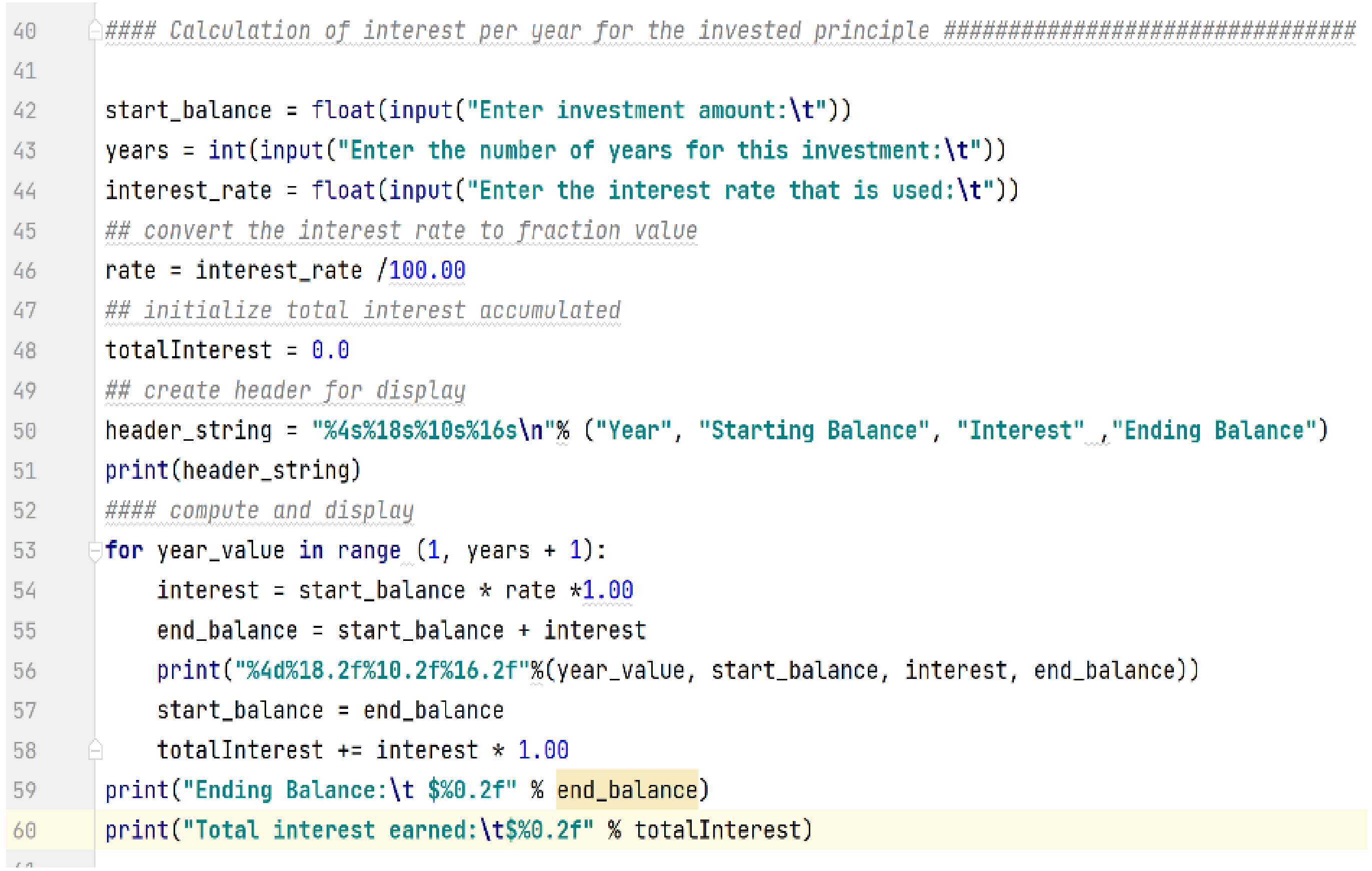
```
C:\Python39\python.exe C:/Users/muham/Desktop/ApplicationProgrammingWin2021/L
Enter integer value:    -10
Enter integer value:    92
Enter integer value:    98
Enter integer value:    20
Enter integer value:    -5
Enter integer value:    76
Enter integer value:    -34
Enter integer value:    12
Enter integer value:    -76
Enter integer value:    45
Enter integer value:    -59
Enter integer value:    25
Enter integer value:    -100
Enter integer value:    -101
Total numbers entered:   13
Total positive numbers entered: 7
Total negative numbers entered: 6
Sum of all numbers, including positive numbers and negative numbers:    84
Sum of all positive numbers:    368
Sum of all negative numbers:    -284

Process finished with exit code 0
```

Calculate annualized interest amount

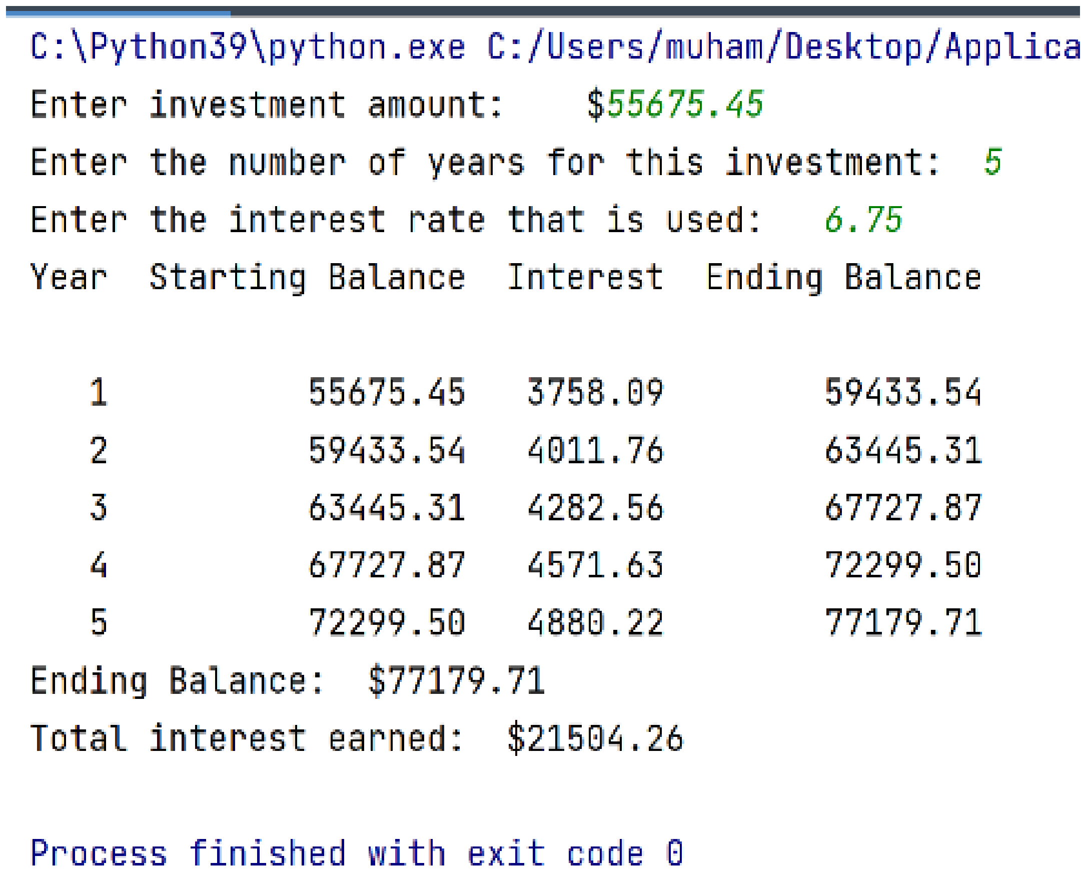


```
"""

Program file name: Lecture2_loop.py
Author: MK
date:   January 15th, 2021
Generate Investment Report
1.  Input of the application are:
    start investment amount (starting amount)
    number of years for which this investment is sought
    interest rate (in terms of %)
2.  Report is displayed in tabular format with heading
3.  Computations and output
    for each year:
        compute interest amount and add it to the investment
        print formatted row of results for that year
4.  Ending investment amount and total interest earned are displayed

"""
```

Calculate interest amountt

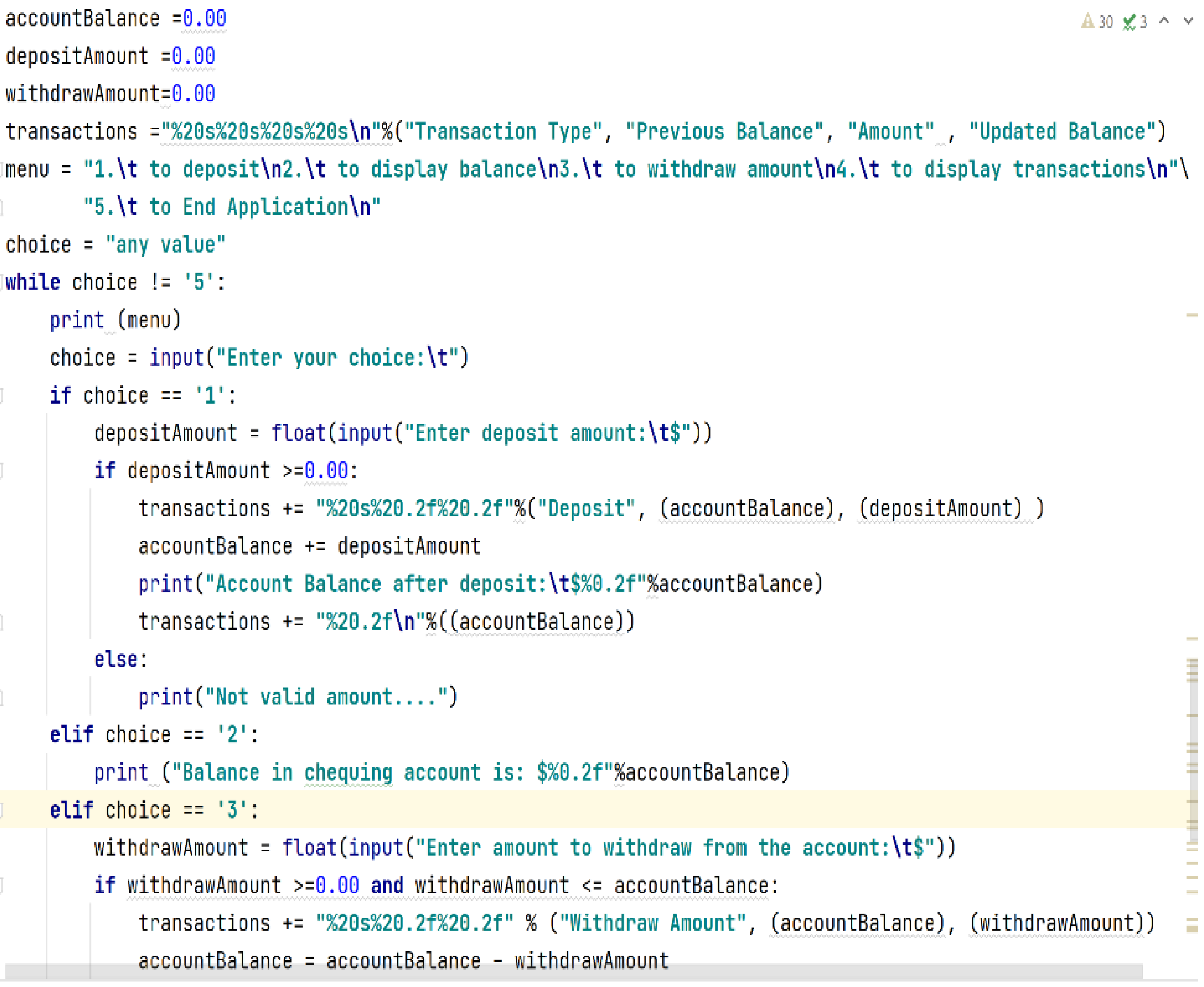


```python
#### Calculation of interest per year for the invested principle ################################

start_balance = float(input("Enter investment amount:\t"))
years = int(input("Enter the number of years for this investment:\t"))
interest_rate = float(input("Enter the interest rate that is used:\t"))
## convert the interest rate to fraction value
rate = interest_rate /100.00
## initialize total interest accumulated
totalInterest = 0.0
## create header for display
header_string = "%4s%18s%10s%16s\n"% ("Year", "Starting Balance", "Interest" ,"Ending Balance")
print(header_string)
#### compute and display
for year_value in range (1, years + 1):
    interest = start_balance * rate *1.00
    end_balance = start_balance + interest
    print("%4d%18.2f%10.2f%16.2f"%(year_value, start_balance, interest, end_balance))
    start_balance = end_balance
    totalInterest += interest * 1.00
print("Ending Balance:\t $%0.2f" % end_balance)
print("Total interest earned:\t$%0.2f" % totalInterest)
```

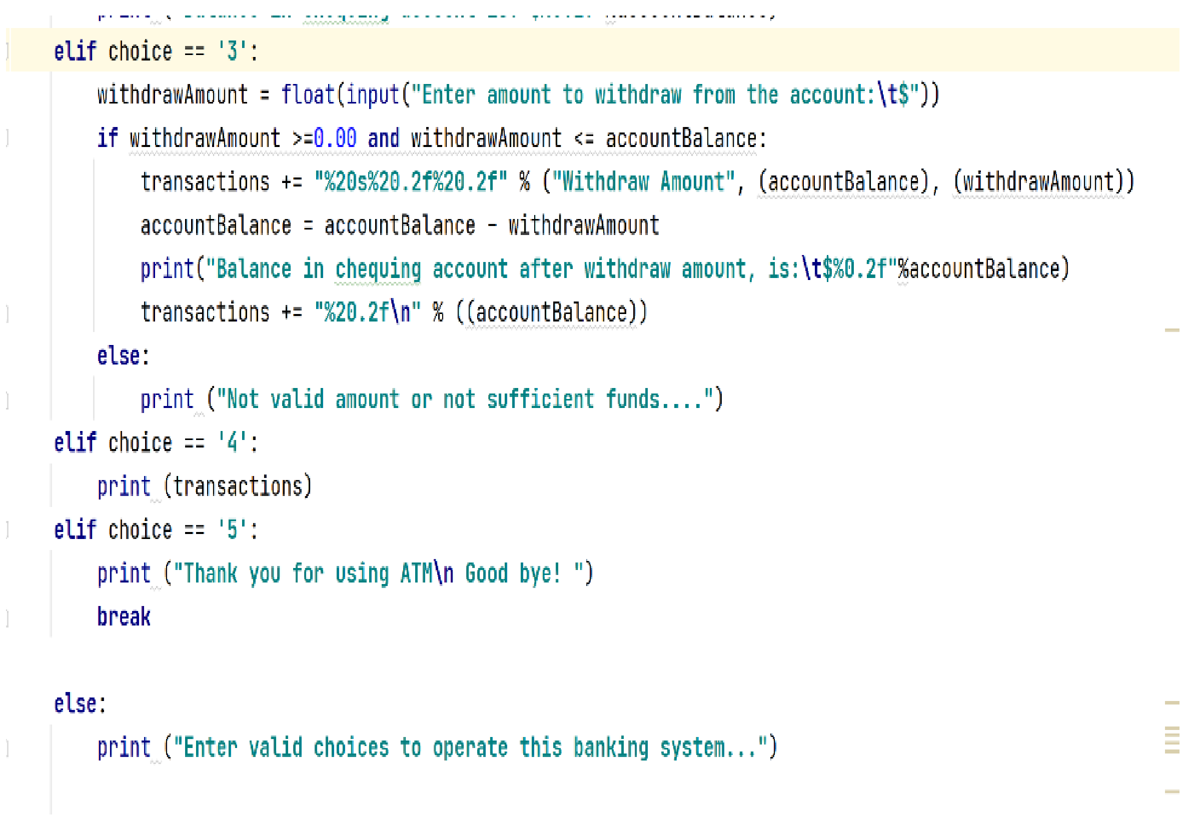


```
C:\Python39\python.exe C:/Users/muham/Desktop/Applica
Enter investment amount:    $55675.45
Enter the number of years for this investment:  5
Enter the interest rate that is used:   6.75
Year  Starting Balance  Interest  Ending Balance


   1          55675.45   3758.09        59433.54
   2          59433.54   4011.76        63445.31
   3          63445.31   4282.56        67727.87
   4          67727.87   4571.63        72299.50
   5          72299.50   4880.22        77179.71
Ending Balance:  $77179.71
Total interest earned:  $21504.26

Process finished with exit code 0
```

My Bank ATM Application

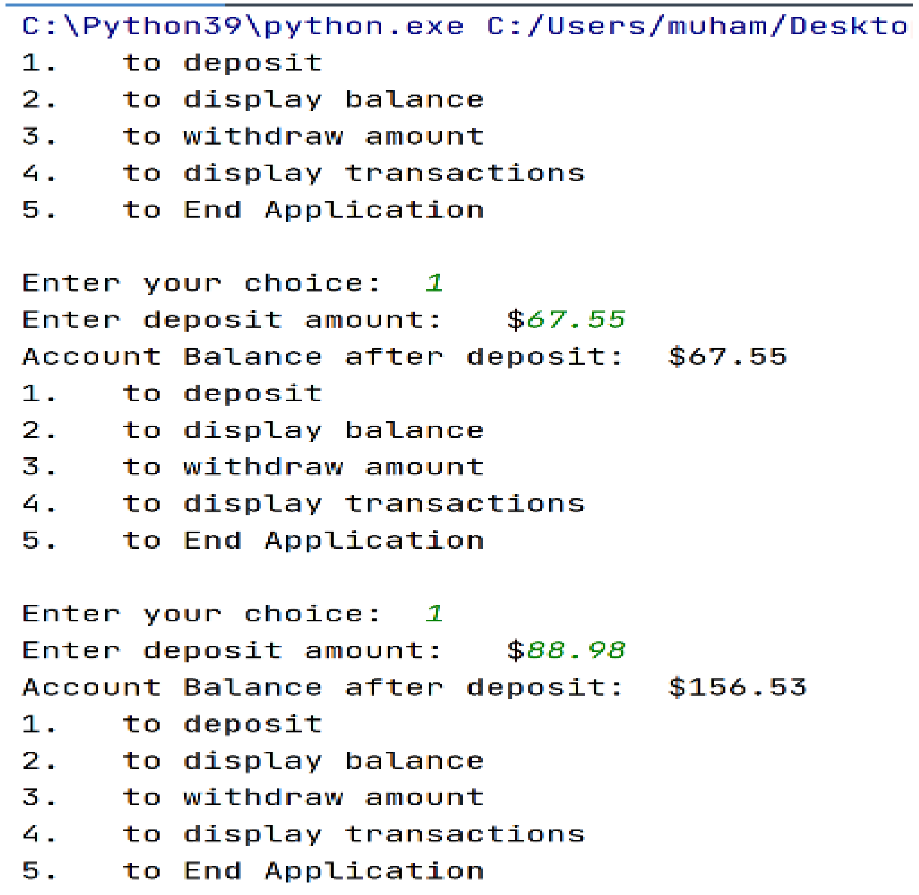


```
accountBalance =0.00
depositAmount =0.00
withdrawAmount=0.00
transactions ="%20s%20s%20s%20s\n"%("Transaction Type", "Previous Balance", "Amount" , "Updated Balance")
menu = "1.\t to deposit\n2.\t to display balance\n3.\t to withdraw amount\n4.\t to display transactions\n"\
       "5.\t to End Application\n"
choice = "any value"
while choice != '5':
    print (menu)
    choice = input("Enter your choice:\t")
    if choice == '1':
        depositAmount = float(input("Enter deposit amount:\t$"))
        if depositAmount >=0.00:
            transactions += "%20s%20.2f%20.2f"%("Deposit", (accountBalance), (depositAmount) )
            accountBalance += depositAmount
            print("Account Balance after deposit:\t$%0.2f"%accountBalance)
            transactions += "%20.2f\n"%((accountBalance))
        else:
            print("Not valid amount....")
    elif choice == '2':
        print ("Balance in chequing account is: $%0.2f"%accountBalance)
    elif choice == '3':
        withdrawAmount = float(input("Enter amount to withdraw from the account:\t$"))
        if withdrawAmount >=0.00 and withdrawAmount <= accountBalance:
            transactions += "%20s%20.2f%20.2f" % ("Withdraw Amount", (accountBalance), (withdrawAmount))
            accountBalance = accountBalance - withdrawAmount
```

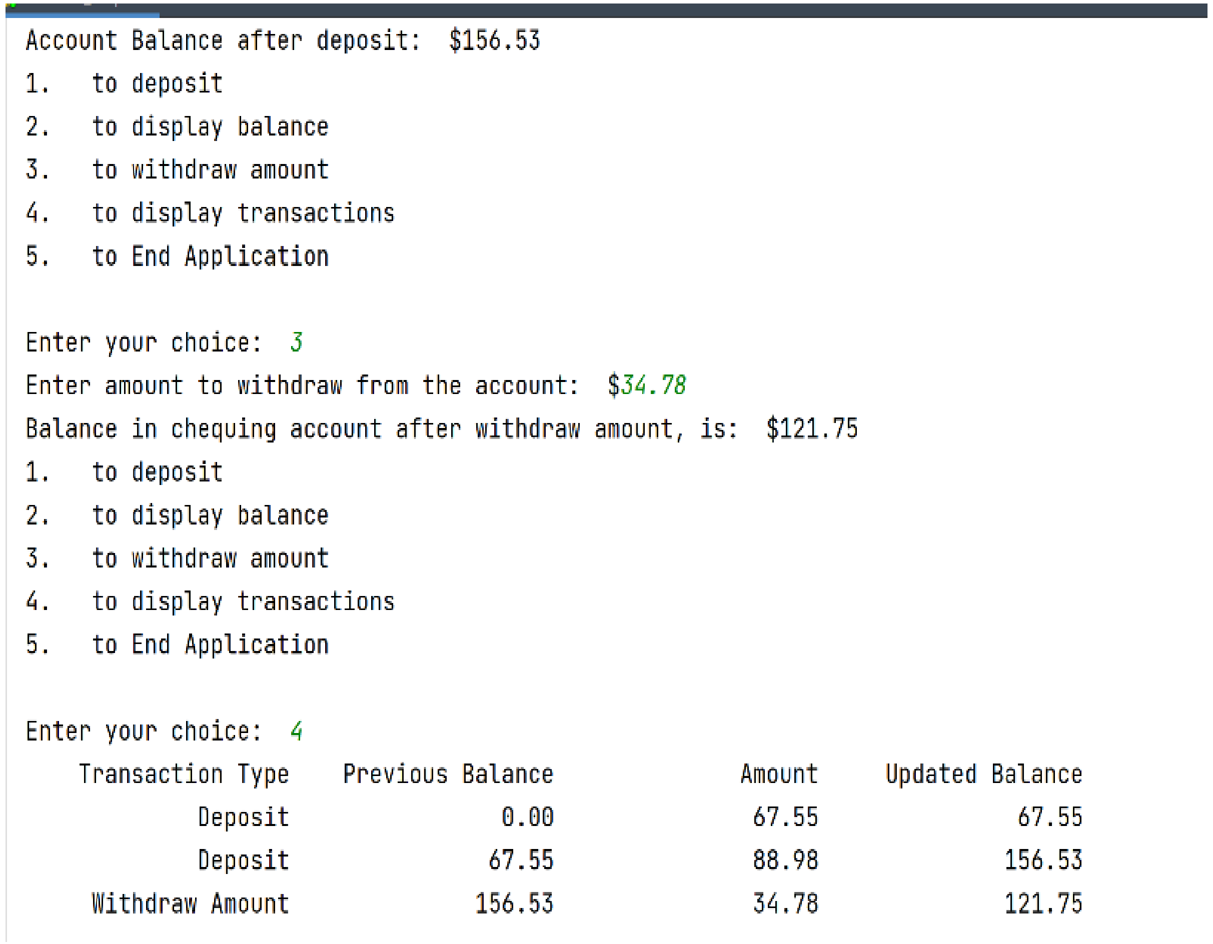


```
    elif choice == '3':
        withdrawAmount = float(input("Enter amount to withdraw from the account:\t$"))
        if withdrawAmount >=0.00 and withdrawAmount <= accountBalance:
            transactions += "%20s%20.2f%20.2f" % ("Withdraw Amount", (accountBalance), (withdrawAmount))
            accountBalance = accountBalance - withdrawAmount
            print("Balance in chequing account after withdraw amount, is:\t$%0.2f"%accountBalance)
            transactions += "%20.2f\n" % ((accountBalance))
        else:
            print ("Not valid amount or not sufficient funds....")
    elif choice == '4':
        print (transactions)
    elif choice == '5':
        print ("Thank you for using ATM\n Good bye! ")
        break

    else:
        print ("Enter valid choices to operate this banking system...")
```

Output of my ATM Bank Application

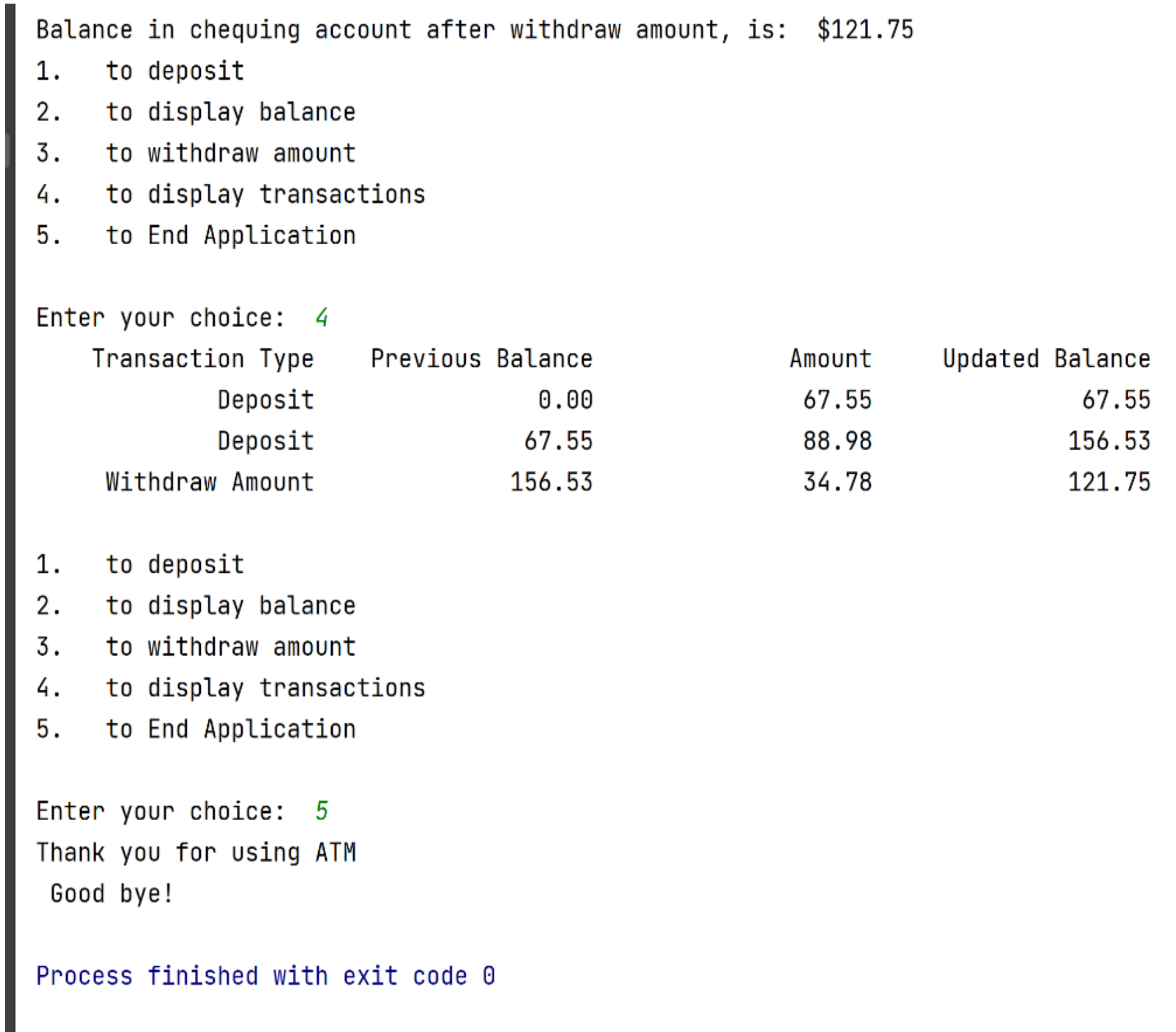


```
C:\Python39\python.exe C:/Users/muham/Deskto
1.    to deposit
2.    to display balance
3.    to withdraw amount
4.    to display transactions
5.    to End Application

Enter your choice:  1
Enter deposit amount:   $67.55
Account Balance after deposit:  $67.55
1.    to deposit
2.    to display balance
3.    to withdraw amount
4.    to display transactions
5.    to End Application

Enter your choice:  1
Enter deposit amount:   $88.98
Account Balance after deposit:  $156.53
1.    to deposit
2.    to display balance
3.    to withdraw amount
4.    to display transactions
5.    to End Application
```



```
Account Balance after deposit:  $156.53
1.    to deposit
2.    to display balance
3.    to withdraw amount
4.    to display transactions
5.    to End Application

Enter your choice:  3
Enter amount to withdraw from the account:  $34.78
Balance in chequing account after withdraw amount, is:  $121.75
1.    to deposit
2.    to display balance
3.    to withdraw amount
4.    to display transactions
5.    to End Application

Enter your choice:  4
    Transaction Type    Previous Balance            Amount     Updated Balance
             Deposit                0.00             67.55               67.55
             Deposit               67.55             88.98              156.53
     Withdraw Amount              156.53             34.78              121.75
```

Output of my bank ATM:



```
Balance in chequing account after withdraw amount, is:  $121.75
1.    to deposit
2.    to display balance
3.    to withdraw amount
4.    to display transactions
5.    to End Application

Enter your choice:  4
    Transaction Type    Previous Balance             Amount     Updated Balance
             Deposit                0.00              67.55               67.55
             Deposit               67.55              88.98              156.53
     Withdraw Amount              156.53              34.78              121.75

1.    to deposit
2.    to display balance
3.    to withdraw amount
4.    to display transactions
5.    to End Application

Enter your choice:  5
Thank you for using ATM
 Good bye!

Process finished with exit code 0
```

Summary of topics covered:

- Control statements
- Control loops with fixed range of values
- Control loops with start and end boundaries
- Control loops with start and end boundaries and defined step size
- Control loop that will have start and end boundaries and step size is defined with negative (reverse)
- Control loop with conditions defined
- Control loops with defined condition to break the loop
- Conditional statements (and multiple conditions)