

Practice exercises: Week 5 (Procedures and Functions)

1. Write a *procedure* called “**toCapitalLetters(s)**” that accepts any English alphabetical text as input (argument) and print its upper-case form. The sample run is here.

```
5 #main Program
6 toCapitalLetters("Ashok kumar")
7 toCapitalLetters("LUT university")
8 string = input("Enter any string:")
9 toCapitalLetters(string)
10
```

Python 3.7.9 (bundled)
->> %Run capitaln_p.py

ASHOK KUMAR
LUT UNIVERSITY
Enter any string:hebut UniVersity
HEBUT UNIVERSITY

Answer:

```
capitaln_p.py x
```

```
1 def toCapitalLetters(s) :
2     print(s.upper())
3
4
5 #main Program
6 toCapitalLetters("Ashok kumar")
7 toCapitalLetters("LUT university")
8 string = input("Enter any string:")
9 toCapitalLetters(string)
10
```

2. Write a *function* called **PoundsToKgs()** that accept pounds as an argument and return in its equivalent kilograms. (One pound is --? refer Weekly Programming Assignment – Week 3)). The sample run is here:

```

4 #main program
5 pounds = float(input("How many pounds are you:"))
6 kg = PoundsToKgs(pounds)
7 print ("Your weight in kgs is:", kg)

```

Shell ×

```

Python 3.7.9 (bundled)|
>>> %Run poundsn_p.py

How many pounds are you:80
Your weight in kgs is: 36.287389600000004

>>> %Run poundsn_p.py

How many pounds are you:120.50
Your weight in kgs is: 54.657880585

```

Answer:

```

1 def PoundsToKgs(p):
2     return p *0.45359237
3
4 #main program
5 pounds = float(input("How many pounds are you:"))
6 kg = PoundsToKgs(pounds)
7 print ("Your weight in kgs is:", kg)
8

```

3. Write a *function* called **cafeMenu()** that prints the list of coffee items and followed by asks input to user to select one of those as an order. The main program (partially hidden here) should print the option selected by the user (you may use if statement). The sample run is here:

```

10
11 #main program
12 x = cafeMenu()

```

Shell ×

```

XYZ cafe_LUT_Lahti
1. Black coffee
2. Coffee with Milk
3. Chocolate Coffee
4. Cappuccino
5. No order exit
Enter your order number:4
you have ordered Cappuccino

>>> %Run coffen_p.py

XYZ cafe_LUT_Lahti
1. Black coffee
2. Coffee with Milk
3. Chocolate Coffee
4. Cappuccino
5. No order exit
Enter your order number:2
you have ordered Coffee with Milk

```

Answer:

```
1 def cafeMenu() :
2     print("XYZ cafe_LUT_Lahti")
3     print("1. Black coffee")
4     print("2. Coffee with Milk")
5     print("3. Chocolate Coffee")
6     print("4. Cappuccino")
7     print("5. No order exit")
8     option = int(input("Enter your order number:"))
9     return option
10
11 #main program
12 x = cafeMenu()
13 if x == 1:
14     print ("you have ordered Black coffee")
15
16 elif x == 2:
17     print ("you have ordered Coffee with Milk")
18
19 elif x == 3:
20     print ("you have ordered Chocolate Coffee")
21
22 elif x == 4:
23     print ("you have ordered Cappuccino")
24
25 else:
26     print ("Thanks and welcome again")
27
28
29
```

4. Write a *procedure* called “**Is_p_n_z()**” with n as parameter that accepts any number (n) as an argument (input) to print whether the given n is positive or negative or zero. The sample run is here:

```
9 #Main program
10 Is_p_n_z(45)
11 Is_p_n_z(-5)
12 Is_p_n_z(0)
```

Shell ×

Python 3.7.9 (bundled)

```
>>> %Run ex1_p.py
It is Positive
It is Negative
It is Zero
>>>
```

Answer

```
1 def Is_p_n_z(n):
2     if n>0:
3         print("It is Positive")
4     elif n==0:
5         print("It is Zero")
6     else:
7         print("It is Negative")
8
9 #Main program
10 Is_p_n_z(45)
11 Is_p_n_z(-5)
12 Is_p_n_z(0)
13
14
```

5. The program written below → To find the greatest number of three given numbers. But it has some decision errors. Create a *subprogram* called **"BiggestOfThree()"** that with three parameters + rewritten If statements to return the correct biggest number when it is called in the main program.

```
1 |
2 x,y,z = 12,-45,-4
3 if x>y>z:
4     print(x)
5 elif y>z:
6     print(y)
7 else:
8     print(z)
9
```

The sample run is here in the #main program

```
print(BiggestOfThree(34,-4,8)) → 34
print(BiggestOfThree(4,-4,8)) → 8
print(BiggestOfThree(-41,-4,-8)) → -4
```

```
a = int(input("enter the first number:"))
b = int(input("enter the first number:"))
c = int(input("enter the first number:"))
print(BiggestOfThree(a,b,c)) → ?
```

Answer:

```
1 def BiggestOfThree(x,y,z):
2     if x>y and x>z:
3         return x
4     elif y>z:
5         return y
6     else:
7         return z
8
9 #main program
10 print(BiggestOfThree(34,-4,8))
11 print(BiggestOfThree(4,-4,8))
12 print(BiggestOfThree(-41,-4,-8))
13
```

6. Write a *function* called **yourGrade()** that accepts your final score (which may be integer or float) as an argument and return the computed grade to the called program. The grade computation table is given here.

Final score	Grade
0-49	0
50-59	1
60-69	2
70-79	3
80-92	4
93-100	5

The sample run is here:

```
17 #main Program
18 print(yourGrade(75.5))
19 print(yourGrade(45))
20 print(yourGrade(69))
21 print(yourGrade(94))
22 print(yourGrade(55))
--
Shell x
Python 3.7.9 (bundled)
>>> %Run grade_p.py
3
0
2
5
1
```

```

1  def yourGrade(s):
2      g=0
3      if s<50:
4          g = 0
5      elif s<60:
6          g = 1
7      elif s<70:
8          g = 2
9      elif s<80:
10         g = 3
11        elif s<92:
12            g = 4
13        else:
14            g = 5
15        return g
16
17    #main Program
18    print(yourGrade(75.5))
19    print(yourGrade(45))
20    print(yourGrade(69))
21    print(yourGrade(94))
22    print(yourGrade(55))
23
24

```

7. Write a function called **powerN()** that accepts any integer with its power value as arguments (n, y) and return the N's power value. That is N^y . You must not use Python's predefined function pow(n, y) or ** operator to get the power of N here. You may need a loop to define this function. The sample run is here.

```

6  #main Program
7  print(powerN(5,2))
8  print(powerN(3,4))
9  print(powerN(2,5))
10
hell x
python 3.7.9 (bundled)
>> %Run power_n_p.py
25
81
32

```

Answer:

```

power_n_p.py x
1  def powerN(x,y):
2      p = 1
3      for i in range(y):
4          p = p*x
5      return p
6  #main Program
7  print(powerN(5,2))
8  print(powerN(3,4))
9  print(powerN(2,5))
10

```

8. Write two subprograms namely **celToFah(c)** and **fahToCel(f)** that convert Celsius to Fahrenheit (procedure) or Fahrenheit to Celsius (function)

respectively. The main program should ask the user to select one of the subprograms that user want to use it to proceed further. The formula for the conversion:

Celsius to Fahrenheit = $(9.0 / 5) * \text{Celsius} + 32$

Fahrenheit to Celsius = $(5 / 9.0) * \text{Fahrenheit} - 32$

The sample run is here:

```
Shell x
Python 3.7.9 (bundled)
>>> %Run fahresn_p.py

select 1 for C to F; OR any number for F to C:1
Enter the Celsius value:35
Fahrenheit = 95.0

>>> %Run fahresn_p.py

select 1 for C to F; OR any number for F to C:3
Enter the Fahrenheit value:120
Celsius = 48.88888888888889
```

Answer:

```
power_n_p.py fahresn_p.py x
1 def celToFah(c):
2     print("Fahrenheit = ",(9.0 / 5) * c + 32)
3
4 def fahToCel(f):
5     return (5 / 9.0) * (f - 32)
6
7 #main program
8 x = int(input(" select 1 for C to F; OR any number for F to C:"))
9
10 if x ==1:
11     x = float(input(" Enter the Celsius value:"))
12     celToFah(x)
13 else:
14     x = float(input(" Enter the Fahrenheit value:"))
15     print("Celsius = ",fahToCel(x))
16
17
18
```

9. Rewrite the exercises 5 and 6 given in *Practice exercises_week 4* as procedure and function respectively. [**Answer Not given for peers check contact your peers to get answer for this.**]

