Paper Coding Worksheet

Chapter:	03
Unit:	17
Student Name:	Aman Dhaubanjar

Q1. Define a function named my_greet that prints "Welcome." and call this function twice to print this greeting

Solution:

```
def my_greet():
    print("Welcome.")
my_greet()
my_greet()
```

Output:

```
Welcome.
Welcome.
```

Q2. Implement the max2(m, n) function, which takes two parameters named m and n, and returns the larger of these two values, and the min2(m, n) which also takes two parameters named m and n and returns the smaller of these two values. Assign 100 and 200 as arguments and call two functions to check the results.

Solution:

```
def max2(m, n):
    return max(m, n)


def min2(m, n):
    return min(m, n)


m, n = 100, 200
print(f"The greater of {m} or {n} is: {max2(m, n)}")
print(f"The smaller of {m} or {n} is: {min2(m, n)}")
```

Output:

```
The greater of 100 or 200 is: 200
The smaller of 100 or 200 is: 100
```

Q3. We want to change the value of the mile, the unit mainly used in the United States, to the value of the kilometer, the international standard unit. Implement the mile2km(mi) function that takes a mile value as a parameter and returns it in kilometers and calls this function to output 1 to 5 miles as kilometers. In this case, use for - in range to make it repeatable. (Define 1 mile as 1.61 km.)

Solution:

```
def mile2km(mi):
    return mi * 1.61
for i in range(1, 6):
    print(f"{i} mile(s) = {round(mile2km(i), 2)} kilometers")
```

Output:

```
1 mile(s) = 1.61 kilometers
2 mile(s) = 3.22 kilometers
3 mile(s) = 4.83 kilometers
4 mile(s) = 6.44 kilometers
5 mile(s) = 8.05 kilometers
```

Q4. Implement the cel2fah(cel) function that takes a temperature in Celsius (Celsius) as a parameter and returns it in Fahrenheit. Then, call this function to change from 10 to 50 degrees Celsius in units of 10 degrees, and output it in Fahrenheit temperature as the following result.

Solution:

```
def cel2fah(cel):
    return cel * 9 / 5 + 32
for c in range(10, 51, 10):
    print(f"{c} degrees Celsius = {cel2fah(c):.1f} degrees Fahrenheit")
```

Output:

```
10 degrees Celsius = 50.0 degrees Fahrenheit
20 degrees Celsius = 68.0 degrees Fahrenheit
30 degrees Celsius = 86.0 degrees Fahrenheit
40 degrees Celsius = 104.0 degrees Fahrenheit
50 degrees Celsius = 122.0 degrees Fahrenheit
```

Paper Coding Worksheet

Chapter:	03
Unit:	18
Student Name:	Aman Dhaubanjar

Q1. Let's take a number n as input and find the sum from 1 to n. Write this function using a recursive function call.

Solution:

```
def sum_recursive(n):
    if n == 1:
        return 1
    return n + sum_recursive(n - 1)


n = int(input("Enter a number: "))

print(f"The sum from 1 to {n} is: {sum_recursive(n)}")
```

Output:

```
The sum from 1 to 10 is: 55
```

Q2. Python has ** operator, which indicates a square power. However, let's take x and n as arguments and without using operators use a recursive function to compute the x power nth. let's try to output 2 power nth by putting the value of x as 2 and value of n as 10.

Solution:

```
def power(x, n):

    if n == 0:

        return 1

    return x * power(x, n - 1)


x = int(input("Enter the base (x): "))

n = int(input("Enter the exponent (n): "))

print(f"{x} to the power of {n} is: {power(x, n)}")power(x,n)
```

Output:

```
2 to the power of 10 is: 1024
```

Paper Coding Worksheet

Chapter:	03

Unit:	19
Student Name:	Aman Dhaubanjar

Q1. There is a list with integer element values called n_list = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]. Return even_list which only contains items of even number values from n_list by using the filter function and the lambda function.

Solution:

```
n_list=[1,2,3,4,5,6,7,8,9,10]

even_list=[]

for x in filter(lambda x: x%2==0 ,n_list):

        even_list.append(x)

print(f"Even list: {even_list}")
```

Output:

```
Even list: [2, 4, 6, 8, 10]
```

Q2. There is a list with integer unit values called n_list = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]. Return even_list which only contains items of even number values from n_list by using a lambda function. This time, do not use for statement, and instead use list function.

Solution:

```
n_list=[1,2,3,4,5,6,7,8,9,10]

even = filter(lambda x: x%2==0 ,n_list)

print(f"Even list: {list(even)}")
```

Output:

```
Even list: [2, 4, 6, 8, 10]
```

Q3. Write a map function that converts a_list which contains lowercase alphabets like ['a', 'b', 'c', 'd'] to a upper_a_list which contains upper case alphabets like ['A', 'B', 'C', 'D'].

Also, define a function named to_upper that receives lowercase letters as parameters and returns upper- case letters, and convert those lowercase letters.

Solution:

```
def to_upper(char):
    return char.upper()


a_list = ['a', 'b', 'c', 'd']
upper_a_list = list(map(to_upper, a_list))
print(f"Uppercase list: {upper_a_list}")
```

Output:

```
Uppercase list: ['A', 'B', 'C', 'D']
```

Q4. Compute the sum of integers from 1 to 100 by using reduce function and lambda expression inside it. Use range(1,101) as input.

Solution:

```
from functools import reduce
result = reduce(lambda x, y: x + y, range(1, 101))
print(f"Sum of integers from 1 to 100: {result}")
```

Output:

```
Sum of integers from 1 to 100: 5050
```

Paper Coding Worksheet

Chapter:	03
Unit:	20

Student Name:	Aman Dhaubanjar

Q1. Create a nested-function by defining a function named greetings and another function named say_hi inside that function. Call say_hi function within greetings. Then call greetings and print ‘hello’. say_hi function is Q1**.** shown below.

Solution:

```
def greetings():
    def say_hi():
        print("hello")
    say_hi()
greetings()
```

Output:

```
hello
```

Q2. Write the following function calc and assign calc to variable num.Then, execute num(3). Make the execution result 14 as follows.

Solution:

```
def calc():
          a=3
          b=5
          def mul_add(x):
                    return a*x+b
          return mul_add
num= calc()
num(3)
```

Output:

```
14
```

Q3. Build mul_add, the inner function of the nested function calc from the previous problem, by using lambda expressions, and print the following result.

Solution:

```
def calc():
    a=3
    b=5
    return (lambda x: a*x +b)
num=calc()
num(3)
```

Output:

```
14
```

Paper Coding Worksheet

Chapter:	03
Unit:	21
Student Name:	Aman Dhaubanjar

Q1. Construct class Dog and its objects with the functionalities described below.

a) method named def bark(self): . This method prints a barking sound.

b) Generates an instance named Dog and refers my_dog by a command named my_dog=Dog.

c) Prints a barking sound with a method named my_dog.bark()

Solution:

```
class Dog:
    def bark(self):
        print("woof woof")
my_dog=Dog()
my_dog.bark()
```

Output:

```
woof woof
```

Q2. Define class Dog with the functionalities described below and call instances and methods.

a) This class Dog has an attribute named name.

b) Has an initialize method named def __init__(self, name): . This method initializes Dog's name.

c) Has a method named def bark(self). This method prints a barking sound.

d) Generates a my_dog instance that has name 'Bingo' with the command my_dog=Dog('Bingo')

e) Prints the following barking sound with the method my_dog.bark()

"Bingo : woof woof"

Solution:

```
class Dog:
    def __init__(self, name):
        self.name = name
    def bark(self):
        print(f"{self.name}: woof woof")
my_dog = Dog("Bingo")
my_dog.bark()
```

Output:

```
Bingo: woof woof
```