Experiment No. 02

Student Name :_____ **PRN**_____

Course Teacher: **Mrs. Priyanka Gadade, Government College of Engg., Jalgaon**

Objective:

1. Learn about functions - definition, default arguments, multiple return values, variable arguments.
2. Learn Python's data structures - lists, dictionaries, and tuples in detail.

Theory:

2.1 Built-in functions

2.1.1 Type conversion

E.g:

1. >>>int(5.5)
   5 #Output

2. >>>str(67)
   '67' #Output

3. >>>print ('Python version' + 2.7)
   TypeError: cannot concatenate 'str' and 'float' objects #Output

4. >>>print('Python version' + str(2.7))
   Python version2.7 #Output

2.2 User Defined Functions

Write a function that accepts a text and a character as argument and returns the no. of occurrences of the character in the text.

```
def count(text, ch):
    n = 0
    for i in text:
        if ch == i:
            n += 1
    return n

# Test
print count("harishgadade", "a")
```

```
#output
3
```

2.2.1 Default argument values

Modify the function in the previous task, such that the caller can specify whether case should be ignored.

```
The default is to ignore case.
def count(text, ch, ignore_case=True):
    if ignore_case:
        text = text.lower()
        ch = ch.lower()
    n = 0
    for t in text:
        if ch == t:
            n += 1
    return n

# Test
print count("harishgadAde", "A", False)
print count("harishgadade", "A", True)
print count("harishgadade", "A")

#Output
1
3
3
```

2.2.2 Multiple return values

Write a function that returns the smallest and largest element in a list.

```
def min_max(numbers):
    smallest = largest = numbers[0]
    for item in numbers:
        if item > largest:
            largest = item
        elif item < smallest:
            smallest = item
    return smallest, largest
```

```
# Test
smallest, largest = min_max([1, 2, 7, 6, 3, 1, 2, 8, 4])
```

2.2.3 Tuples

Color is represented using three bytes, one each for red, green and blue. For the color with components red = 240, green = 50 and blue = 150.

1. Represent the color as a tuple.
2. Show that tuples are immutable.
3. Represent the color as a tuples using the alternate syntax (without parenthesis).
4. Extract the components into separate variable for red, green, and blue.

```
>>> rgb = (240, 50, 150)
>>>
>>> rgb[0] = 150
        Traceback (most recent call last):
        File "<stdin>", line 1, in <module>
        TypeError: 'tuple' object does not support item assignment
>>>
>>> rgb = 240, 50, 150
>>> print rgb
(240, 50, 150)
>>>
>>> r, g, b = rgb
>>> print r
240
>>> print g
50
>>> print b
150
>>>
```

2.2.4 Dictionaries

A table of names and marks are given below.

Marks	**Name**
4098	harish

4139	vihaan

1. Represent the table as dictionary mapping names to marks.
2. Add a new entry that maps chetana to 4127.
3. Get the marks of harish, from the dictionary.
4. Remove the entry for vihaan from the dictionary.
5. Get the names from the dictionary.
6. Get the marks from the dictionary.
7. Check if chetana is in the dictionary.

```
>>> tel = {'harish': 4098, 'vihaan': 4139}
>>>
>>> tel['chetana'] = 4127
>>> tel == {'vihaan': 4139, 'chetana': 4127, 'harish': 4098}
True
>>>
>>> print tel['harish']
4098
>>>
>>> tel.pop('vihaan')
4139
>>> tel == {'chetana': 4127, 'harish': 4098}
True
>>>
>>> print sorted(tel.keys())
['chetana', 'harish']
>>>
>>> print sorted(tel.values())
[4098, 4127]
>>>
>>> print 'chetana' in tel
True
>>>
```

Lab Practices:

Write a menu driven program for above examples.

Course Teacher Sign and Date