Python Assignment 3

2282437

Zaid Shariff

(1) Demonstrate the following operations of Python Functions with suitable scenario of your choice

```
# (a) Creating a function with parameters
# Function find sum of first even or odd numbers, depending upon the input. If even number
def eve_odd(n):
  sum=0
  if n%2==0:
    for i in range(2,n+1,2):
      sum+=i
    print(sum)
  else:
    for i in range(1,n+1,2):
      sum+=i
    print(sum)
```

```
eve_odd(45)
```

```
529
```

```
# (b) Creating a function with parameters and return value
# Input a given number of number, the function will sort it in ascending order and return
def lrgst(l):
  l.sort()
  return(l,max(l))
```

```
lrgst([25,53,36,88,22,-69,63,188])
```

```
([-69, 22, 25, 36, 53, 63, 88, 188], 188)
```

```
# (c) Calling a function
# A function returns true if a given number is a trianguler number, else returns false
def trangular():
  n=int(input("Enter a number n "))
  if n<=0:
    return False
  else:
    sum,i=0,1
    while(sum<=n):
      sum+=i
      if (sum==n):
```

```
        return True
      i+=1
    return False
```

```
if trangular():
  print("The entered number is a triangular number")
else:
    print("The entered number is not a triangular number")
```

```
Enter a number n 10
The entered number is a triangular number
```

```
# (d) Calling a function of a module
# Function that returns the squarer root of a given number
import math

inpt=int(input("Enter a number n "))
sqt=math.sqrt(inpt)
print(sqt)
```

```
Enter a number n 45
6.708203932499369
```

```
# (e) Variable-length Arguments
# Function takes a list and returns the list in a sorted way.
def fnct(*args):
  l=[]
  for i in args:
    l.append(i)
    l.sort()
  return l
```

```
fnct(11,102,35553,88,55)
```

```
[11, 55, 88, 102, 35553]
```

```
# (f) Recursive function
# Sum of first n numbers

def factorial(n):
    if (n==1 or n==0):
        return 1
    else:
        return (n * factorial(n - 1))
```

```
factorial(7)
```

```
5040
```

(2) Demonstrate the following Python Tuple operations with suitable examples

```
# (a) Creating and printing Tuples
tpl=(1,2,3,4,"sah",[1,2,3])
print(tpl)
```

```
(1, 2, 3, 4, 'sah', [1, 2, 3])
```

```
# (b) Accessing Tuple items (Positive and Negative Indexing)
tpl[1]
```

```
2
```

```
tpl[-2]
```

```
'sah'
```

```
# (c) Slicing Tuples
tpl[0:5]
```

```
(1, 2, 3, 4, 'sah')
```

```
tpl[-1::-1]
```

```
([1, 2, 3], 'sah', 4, 3, 2, 1)
```

```
# (d) Iterating over Tuple, Tuple Membership Test
# check if 'sah' is present in tpl
if 'sah' in tpl:
  print('''"sah" is present in tpl''')
if 'AAA' in tpl:
  print('''"AAA" is present in tpl''')
```

```
"sah" is present in tpl
```

```
# (e) Concatenating Tuples
t1=(1,2,3,4,56,7,989,"apple","mango")
t2=("Monday","Tuesday","Wednesday")
print(t1+t2)
```

```
(1, 2, 3, 4, 56, 7, 989, 'apple', 'mango', 'Monday', 'Tuesday', 'Wednesday')
```

```
# (f) Converting List into Tuple
lst=[45,25,39,77,58,36,47,58,"HVAH"]
```

```
tpl=tuple(lst)
tpl
```

```
(45, 25, 39, 77, 58, 36, 47, 58, 'HVAH')
```

```
# (g) sorted(), count(), index()
tpl1=(1,23,4,23,23,56,98,69,6)
sorted(tpl1)
```

```
[1, 4, 6, 23, 23, 23, 56, 69, 98]
```

```
tpl1.count(23)
```

```
3
```

```
tpl1.index(23)
```

```
1
```

(3) Demonstrate the following Python Dictionary operations with suitable examples

```
# (a) Creating and printing a dictionary
dct={"India":"Democracy", "Russia":"Socialist", "China":"Communist", "Saudi":"Monarchy"}
print(dct)
```

```
{'India': 'Democracy', 'Russia': 'Socialist', 'China': 'Communist', 'Saudi': 'Monarch
```

```
# (b) Accessing Dictionary [items(), keys() and values()]
dct.items()
```

```
dict_items([('India', 'Democracy'), ('Russia', 'Socialist'), ('China', 'Communist'),
('Saudi', 'Monarchy')])
```

```
dct.values()
```

```
dict_values(['Democracy', 'Socialist', 'Communist', 'Monarchy'])
```

```
dct.keys()
```

```
dict_keys(['India', 'Russia', 'China', 'Saudi'])
```

```
# (c) Changing Item Value and Adding items to Dictionary
dct["China"]="Socialist"
print(dct)
```

```
{'India': 'Democracy', 'Russia': 'Socialist', 'China': 'Socialist', 'Saudi': 'Monarch
```

```
dct["USA"]="Democracy"
print(dct)
```

```
{'India': 'Democracy', 'Russia': 'Socialist', 'China': 'Socialist', 'Saudi': 'Monarch
```

```
# (d) Deleting or removing elements from a Dictionary
dct.pop("India")
print(dct)
```

```
{'Russia': 'Socialist', 'China': 'Socialist', 'Saudi': 'Monarchy', 'USA': 'Democracy
```

```
del dct["Russia"]
print(dct)
```

```
{'China': 'Socialist', 'Saudi': 'Monarchy', 'USA': 'Democracy'}
```

```
# (e) Iterating over Dictionaries
dct={"India":"Democracy", "Russia":"Socialist", "China":"Communist", "Saudi":"Monarchy", "
for key, value in dct.items():
    if value=="Democracy":
       print(key,value)
```

```
India Democracy
USA Democracy
```

```
# (f) update(), len(), sorted(), clear()
dct={"India":"Democracy", "Russia":"Socialist", "China":"Communist", "Saudi":"Monarchy", "
dct.update(England="Monarchy",Japan="Monarcy")
print(dct)
```

```
{'India': 'Democracy', 'Russia': 'Socialist', 'China': 'Communist', 'Saudi': 'Monarch
```

```
len(dct)
```

```
7
```

```
print(sorted(dct.keys()))
```

```
['China', 'England', 'India', 'Japan', 'Russia', 'Saudi', 'USA']
```

```
print(dct)
dct.clear()
print(dct)
```

```
{'India': 'Democracy', 'Russia': 'Socialist', 'China': 'Communist', 'Saudi': 'Monarch
{}
```

✓ 0s completed at 12:28 PM