Assignment

Q1. Which keyword is used to create a function? Create a function to return a list of odd numbers in the range of 1 to 25.

Ans. The keyword **def** is used for defining the functions. The sample function to return the list of odd numbers from 1 to 25:

```
def oddf(n):
    for i in range(1,n):
        if (i%2)!=0:
            yield i
#function defination
for i in oddf(25):
    print(i)
#Calling function
```

OUTPUT:

1

3

5

7

9

11

13

15

17

19

21

23

Q2. Why *args and **kwargs is used in some functions? Create a function each for *args and **kwargs to demonstrate their use.

Ans.

- ***args**

  This keyword is used to accept n number of values in function and the type of the output of the function is a tuple.

  Example:

  ```
  def test1(*args):
      return args

  test1(1,2,3,4,5,67,"Ashish",True)
  ```

  OUTPUT:

  ```
  (1, 2, 3, 4, 5, 67, 'Ashish', True)
  ```

- ****args**

  This keyword is used to ddefine the values in the function in the form of key and values and the type of the output of the function is dictionary.

  Example:

```
def test22(**kwarg):
    return kwarg

test22(a=1, b= "Ashish",c=(1,2,3,4,5),d=[1,2,3,4,5,6,True,"Ashish"],e={1,1,1,2,3,4,"Lucky"})
```

  OUTPUT:

```
{'a': 1,
 'b': 'Ashish',
 'c': (1, 2, 3, 4, 5),
 'd': [1, 2, 3, 4, 5, 6, True, 'Ashish'],
 'e': {1, 2, 3, 4, 'Lucky'}}
```

Q3. What is an iterator in python? Name the method used to initialise the iterator object and the method used for iteration. Use these methods to print the first five elements of the given list [2, 4, 6, 8, 10, 12, 14, 16, 18, 20].

Answer. An iterator is an object that contains a countable number of values. An iterator is an object that can be iterated upon, meaning that you can traverse through all the values.

```
def func(a):
    l = []
    for i in range(0,5):
        l.append(a[i])
    return l
l=[2, 4, 6, 8, 10, 12, 14, 16,18, 20]
func(l)
```

Output:

```
[2, 4, 6, 8, 10]
```

Q4. What is a generator function in python? Why yield keyword is used? Give an example of a generator function.

Answer. Generator function allows to define an iterative algorithm by writing a single function whose execution is not continuous.

The yield keyword pauses generator function execution and the value of the expression following the yield keyword is returned to the generator's caller.

Example:

```
def fib(n):
    a,b=0,1
    for i in range(n):
        yield a
        a,b=b,a+b
```

```
for i in fib(10):
    print(i)
```

Output:

0

1

1

2

3

5

8

13

21

34

Q5. Create a generator function for prime numbers less than 1000. Use the next() method to print the first 20 prime numbers.

Answer:

```
for Number in range (1, 1000):
    count = 0
    for i in range(2, (Number//2 + 1)):
        if(Number % i == 0):
            count = count + 1
            break

    if (count == 0 and Number != 1):
        print(" %d" %Number, end = '  ')
```