***Functions 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.**

**“def” keyword is used to create a function.**

**def odd\_numbers():**

**n = []**

**for i in range(1, 26):**

**if i % 2 != 0:**

**n.append(i)**

**return n**

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

**\*args and \*\*kwargs have special type of syntax in python, each of them have special functioning that’s why it is used in function.**

**\*args is used to pass a variable number of non-keyworded arguments to a function. It allows you to pass any number of positional arguments to a function. These arguments are passed to the function as a tuple.**

**def print\_args(\*args):**

**for arg in args:**

**print(arg)**

**print\_args(1, 2, 3) # Output: 1 2 3**

**print\_args('a', 'b', 'c', 'd') # Output: a b c d**

**\*\*kwargs is used to pass a variable number of keyworded arguments to a function. It allows you to pass any number of named arguments to a function. These arguments are passed to the function as a dictionary.**

**def print\_kwargs(\*\*kwargs):**

**for key, value in kwargs.items():**

**print(key, value)**

**print\_kwargs(a=1, b=2, c=3) # Output: a 1 b 2 c 3**

**print\_kwargs(name='John', age=25, city='New York') # Output: name John age 25 city New York**

**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].**

**In Python, an iterator is an object that can be iterated (looped) upon. It is used to traverse a sequence of values, such as a list or a tuple. The iterator can be used to access each element of the sequence one by one, and it keeps track of the current position within the sequence.**

**To create an iterator object in Python, we use the iter() method. This method takes a sequence as an argument and returns an iterator object.**

**To iterate over the elements of an iterator, we use the next() method. This method retrieves the next element from the iterator and advances the current position. When we have reached the end of the sequence, the next() method raises a StopIteration exception.**

**list = [2, 4, 6, 8, 10, 12, 14, 16, 18, 20]**

**my\_iterator = iter(list)**

**print(next(my\_iterator)) # Output: 2**

**print(next(my\_iterator)) # Output: 4**

**print(next(my\_iterator)) # Output: 6**

**print(next(my\_iterator)) # Output: 8**

**print(next(my\_iterator)) # Output: 10**

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

**A generator function is a special type of function in Python that returns an iterator object which can be used to iterate over a sequence of values. The main difference between a regular function and a generator function is that a regular function returns a single value, whereas a generator function can return a sequence of values, one at a time.**

**The yield keyword is used in generator functions to define a point at which the function should pause its execution and return a value to the caller. When the generator function is called again, it resumes its execution from the point at which it was paused and continues until it reaches the next yield statement.**

**def my\_generator(n):**

**for i in range(n):**

**yield i**

**# Example usage**

**for value in my\_generator(5):**

**print(value)**

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

**def generate\_primes():**

**primes = []**

**for num in range(2, 1000):**

**is\_prime = True**

**for prime in primes:**

**if num % prime == 0:**

**is\_prime = False**

**break**

**if is\_prime:**

**primes.append(num)**

**yield num**

**# Example usage**

**prime\_generator = generate\_primes()**

**for i in range(20):**

**print(next(prime\_generator))**