Functions and Modules

```
def greet(name):
  print("hello, " + name + "!")
greet ("Alice")
```

```
hello, Alice!
```

Function Arguments 1.Positional Arguments

```
def add(a,b):
  return a+b
print(add(5,3))
```

```
8
```

Keyword Argument

```
def greet(name, message):
  print(message + ", " + name + "!")
greet(name="Alice", message="Good morning")
```

```
Good morning, Alice!
```

Default Argument

```
def greet(name, message="hello"):
  print(message + "," + name + "!")
greet("Alice")
greet("Bob", "hi")
```

```
hello,Alice!
hi,Bob!
```

Variable-length Argument

```
def sum_numbers(*numbers):
  return sum(numbers)
print(sum_numbers(1,2,3,4))
```

```
10
```

**kwargs

```
def describe_person(**kwargs):
  for key, value in kwargs.items():
    print(f"{key}: {value}")
describe_person(name="Alice", age=25, city="New york")
```

```
name: Alice
age: 25
city: New york

#variable length argument
#** keyword argument
def Pooja(**numbers):
  for key, value in numbers.items():
        print(f"{key}: {value}")
Pooja(name="Pooja", age=25, city="New york")

name: Pooja
age: 25
city: New york
```

Return statement

```
def square(num):
 return num*num
result=square(5)
print("square is:", result)

square is: 25
```

MODULES

Using Built-in modules

1.Importing Modules

```
import math
print(math.sqrt(16))

4.0
```

Import Specific Functions

```
from math import pi, sin
print(pi)
print(sin(math.radians(90)))

3.141592653589793
1.0
```

Import and use the module in another file

```
def greet(name):
       print("hello, " + name + "!")
def add(a, b):
       return a + b
```

```
 print(my_module.add(5,3))
```

PRIME NUMBER OR NOT

```
def is_prime(num):
  if num <= 1:
    return False

  for i in range(2, int(num**0.5) + 1):
   if num % i == 0:
    return False
    return True
number = int(input("enter a number: "))
if is_prime(number):
  print("the number is prime.")
else:
  print("the number is not prime")

enter a number: 6
the number is not prime
```

Create a function to Generate Fibboacci Sequence

```
def fibonacci(n):
  sequence = []
  a, b = 0, 1
  for _ in range(n):
    sequence.append(a)
    a,b = b, a+b
  return sequence

terms = int(input("enter the number of terms:"))
print("fibbonacci sequence:",fibonacci(terms))

enter the number of terms:10
fibbonacci sequence: [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
```

Use the math module to solve a problem

```
import math

angle = float(input("enter an angle in degrees:"))
radian = math.radians(angle)
print("sin of angle:", math.sin(radian))
print("cosine of angle:", math.cos(radian))
```

```
enter an angle in degrees: 30
sin of angle: 0.49999999999999994
cosine of angle: 0.8660254037844387
```

Factorial Using Recursion

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


num = int(input("enter a number: "))
print("factorial:", factorial(num))
```

```
enter a number: 10
factorial: 3628800
```

Fibbonacci Series Using Functions

```
def fibonacci(n):
  a, b = 0,1
  for _ in range(n):
    print(a, end=" ")
    a,b = b, a+b
count = int(input("enter the number of terms:"))
fibonacci(count)
```

```
enter the number of terms:10
0 1 1 2 3 5 8 13 21 34
```

Reverse a string using a function

```
def reverse_string(s):
  return s[::-1]


text = input("enter a string:")
print("Reversed stirng:", reverse_string(text))
```

```
enter a string:50
Reversed stirng: 05
```

Find GCD of two numbers using a function

```
def gcd(a, b):
  while b:
    a,b = b, a % b
```

```
    return a

num1 = int(input("enter first number: "))
num2 = int(input("enter second number: "))
print("GCD:", gcd(num1 , num2))
```

```
enter first number: 50
enter second number: 40
GCD: 40
```