Module – 2 (Fundamentals of python)

1. Write a Python program to check if a number is positive, negative or zero.
   - num = int(input("Enter a number: "))

```
if num > 0:
   print("The number is positive.")
elif num < 0:
   print("The number is negative.")
else:
   print("The number is zero.")
```

2. Write a Python program to get the Factorial number of given number.
   - num = int(input(“Enter the Number:”))

```
factor = 1

if num < 0:
        print("Sorry, Factorial does not exist for Negative
Numbers")
elif num == 0:
        print("The factorial of 0 is 1")
else:
        for i in range(1,num + 1):
      factorial = factorial*i
print("The factorial of",num,"is",factorial)
```

3. Write a Python program to get the Fibonacci series of given range.

```
a = 1
b = 0
print (b, a, end= " ")
for i in range(0,10):
    c = b
    b = a
    a = c + b
    print(a, end=" ")
```

4. How memory is managed in Python?

- According to the Python memory management documentation, Python has a private heap that stores our program's objects and data structures. Python memory manager takes care of the bulk of the memory management work and allows us to concentrate on our code.
- There are two types of memory allocation in Python, static and dynamic.
  - **Static memory**

    The stack data structure provides **static memory allocation**, meaning the variables are in the stack memory. Statically assigned variables, as the name implies, are permanent; this means that they must be allocated in advance and persist for the duration of the program. Another point to remember is that we cannot reuse the memory allocated in the stack memory. Therefore, memory reusability is not possible.

  - **Dynamic memory**

    The dynamic memory allocation **uses heap data structures in its implementation, implying that variables are in the heap memory. As the name suggests, dynamically allocated variables are not permanent and can be changed while a program is running. Additionally, allotted memory can be relinquished and reused. However, it takes longer to**

complete because dynamic memory allocation occurs during program execution. Furthermore, after utilizing the allocated memory, we must release it. Otherwise, problems such as memory leaks might arise.

- **Memory management in Python involves a private heap containing all Python objects and data structures. The management of this private heap is ensured internally by the Python memory manager. The Python memory manager has different components which deal with various dynamic storage management aspects, like sharing, segmentation, preallocation or caching.**

5. What is the purpose continue statement in python?

- **In Python, the `continue` statement is used inside loops (such as `for` and `while` loops) to skip the rest of the current iteration and move on to the next iteration. When the `continue` statement is encountered, the remaining code inside the loop for the current iteration is skipped, and the loop proceeds to the next iteration, if any.**
- **The purpose of the `continue` statement is to control the flow of the loop and skip certain iterations based on a specific condition. It is often used when you want to skip over certain iterations when a certain condition is met, but you don't want to exit the loop entirely.**
- **for x in range(9):**

  if x == 3:

```
        continue
print(x, end = " ")
```

6. **Write python program that swap two number with temp variable and without temp variable.**

- **# Swapping with a Temporary Variable**
  - **a = 5**

    b = 10

    temp = a

    a = b

    b = temp

    print("After swapping with a temp variable:")

    print("a =", a)

    print("b =", b)
- **# Swapping without a Temporary Variable**
  - a = 15

    b = 20

    a = a + b

    b = a - b

    a = a - b

    print("After swapping without a temp variable:")

    print("A = ", a)

    print("B = ", b)

7. Write a Python program to find whether a given number is even or odd, print out an appropriate message to the user.
   -
```
num = int(input("Enter a number: "))
if num % 2 == 0:
   print(f"{num} is even.")
else:
   print(f"{num} is odd.")
```

8. Write a Python program to test whether a passed letter is a vowel or not.
   -
```
letter = input("Input a letter of the alphabet: ")
if letter in ('a', 'e', 'i', 'o', 'u'):
        print(letter, " is a vowel." )
else:

        print(letter, " is a consonant." )
```

9. • Write a Python program to sum of three given integers. However, if two values are equal sum will be zero.
   -