

# 11<sup>TH</sup> JUNE ASSIGNMENT

1. What is a lambda function in Python, and how does it differ from a regular function?

ANS: Lambda function is an anonymous function that takes many arguments and returns one expression. It differs from normal functions because there is no specific function name and it does not involve parentheses.

2. Can a lambda function in Python have multiple arguments? If yes, how can you define and use them?

ANS: Yes, Lambda function can have multiple arguments. They can be defined as single line functions and used to return a single result.

3. How are lambda functions typically used in Python? Provide an example use case.

ANS: Syntax → lambda arguments : expression

Eg. lambda x : x+5

4. What are the advantages and limitations of lambda functions compared to regular functions in Python?

ANS: Advantages:

- Concise code
- No function name
- Single line expression

Disadvantages:

- Can contain only one line
- Complex problem cannot be handled
- Difficult to understand

5. Are lambda functions in Python able to access variables defined outside of their own scope?

Explain with an example.

ANS: No, lambda functions cannot access variables defined outside the scope.

6. Write a lambda function to calculate the square of a given number.

ANS:

```
num = input(int())  
square = lambda num : num**2  
print (square)
```

7. Create a lambda function to find the maximum value in a list of integers.

ANS:

```
list = [20, 10, 20, 4, 100]
print(max(list, key=lambda value: int(value)) )
```

8. Implement a lambda function to filter out all the even numbers from a list of integers.

ANS:

```
nums = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
even_nums = list(filter(lambda x: x%2 == 0, nums))
print(even_nums)
```

9. Write a lambda function to sort a list of strings in ascending order based on the length of each string.

ANS:

```
myList = ["JothiKrishna", "Gokul Akash", "Janani", "Sruthi", "Hariharan", "Susanna",
          "Sandy"]
sortedList = sorted(myList, key=lambda x: len(x))
print(sortedList)
```

10. Create a lambda function that takes two lists as input and returns a new list containing the common elements between the two lists.

ANS:

```
list1 = [1, 2, 3, 5, 7, 8, 9, 10]
list2 = [1, 2, 4, 8, 9]
result = list(filter(lambda x: x in list1, list2))
print ("\nIntersection of the said list: ",result)
```

11. Write a recursive function to calculate the factorial of a given positive integer.

ANS:

```
def factorial(n):
    if n == 0:
        return 1
    return n * factorial(n-1)
num = input(int());
print("Factorial of", num, "is", factorial(num))
```

12. Implement a recursive function to compute the nth Fibonacci number.

ANS:

```
def Fibonacci(n):  
    if n <= 0:  
        print("Incorrect input")  
    elif n == 1:  
        return 0  
    elif n == 2:  
        return 1  
    else:  
        return Fibonacci(n-1)+Fibonacci(n-2)  
num= input(int())  
print(Fibonacci(num))
```

13. Create a recursive function to find the sum of all the elements in a given list.

ANS:

```
list1 = [1,2,3,4,5]  
def sumOfList(list, size):  
    if (size == 0):  
        return 0  
    else:  
        return list[size - 1] + sumOfList(list, size - 1)  
result = sumOfList(list1, len(list1))  
  
print("Sum of all elements in given list: ", result)
```

14. Write a recursive function to determine whether a given string is a palindrome.

ANS:

```
def isPalindrome(str, i):
    if(i > len(str)/2):
        return True
    ans = False
    if((str[i] is str[len(str) - i - 1]) and isPalindrome(str, i + 1)):
        ans = True
    return ans

intstr = input(str())
if (isPalindrome(intstr, 0)):
    print("Yes")
else:
    print("No")
```

15. Implement a recursive function to find the greatest common divisor (GCD) of two positive

ANS:

```
def hcf(a, b):
    if (b == 0):
        return a
    else:
        return hcf(b, a % b)

a = input(int())
b = input(int())
print("The gcd of a and b is : ", end="")
print(hcf(a, b))
```