Name: Mukesh Kumar Behera

Assignment: Introduction to Python

1.What are the types of Applications?

Ans: Web Applications, GUI Applications, Artificial Intelligence & Machine Learning etc.

2. What is programming?

Ans: Programming is the process of writing logical statements that a computer can understand and follow to perform specific tasks.

3. What is Python?

Ans: Python is a high-level, interpreted, and general-purpose programming language. It is popular because it is simple, powerful, and versatile.

4. Write a Python program to check if a number is positive, negative or zero.

```
   num = float(input("Enter a number: "))

 Ans  if num > 0:
    print("The number is Positive.")
  elif num < 0:
    print("The number is Negative.")
  else:
    print("The number is Zero.")
```

5. Write a Python program to get the Factorial number of given numbers.

```
Ans: num = int(input("Enter a number: "))
      factorial = 1
      for i in range(1, num + 1):
        factorial = factorial * i
        print(f"The factorial of {num} is {factorial}")
```

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

Ans:

```
n = int(input("Enter number of terms: "))
a, b = 0, 1
for i in range(n):
 print(a, end=" ")
 a, b = b, a + b
```

7. How memory is managed in Python?

Ans: *Python handles memory automatically, unlike C/C++ where you manually manage memory.

* Large objects and cyclic references are cleaned by the garbage collector.

*Developers mainly need to avoid unnecessary references to objects to free memory efficiently.

8. What is the purpose continuing statement in python?

Ans: To skip certain conditions in a loop without stopping the entire loop.

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

Ans: * Using temp variable:-

```
a = int(input("Enter first number: "))
b = int(input("Enter second number: "))
temp = a
a = b
b = temp
print("After swapping: a =", a, "b =", b)
```

*Without temp Variable

```
a = int(input("Enter first number: "))
b = int(input("Enter second number: "))
```

```
a, b = b, a    print("After swapping: a =", a, "b =", b)
```

10. Write a Python program to find whether a given number is even or odd, print out an appropriate message to the user.

Ans:

```
num = int(input("Enter a number: "))
if num % 2 == 0:
  print(f"{num} is Even.")
else:
   print(f"{num} is Odd.")
```

11. Write a Python program to test whether a passed letter is a vowel or not.

Ans:

```
letter = input("Enter a letter: ").lower()  # convert to lowercase
 if letter in 'aeiou':
   print(f"{letter} is a vowel.")
else:
  print(f"{letter} is not a vowel.")
```

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

Ans:

```
a = int(input("Enter first number: "))
b = int(input("Enter second number: "))
c = int(input("Enter third number: "))
if a == b or a == c or b == c:
    total = 0
else:
  total = a + b + c
```

```
         print("Result:", total)
```

13. Write a Python program that will return true if the two given integer values are equal or their sum or difference is 5.

Ans:

```
a = int(input("Enter first number: "))
         b = int(input("Enter second number: "))
        if a == b or (a + b) == 5 or abs(a - b) == 5:
           print(True)
       else:
         print(False)
```

14. Write a python program to sum of the first n positive integers.

Ans:

```
n= int(input("Enter a positive integer: "))
                total = n * (n + 1) // 2
                  print(f"The sum of first {n} positive integers is {total}.")
```

15. Write a Python program to calculate the length of a string.

Ans:

```
text = input("Enter a string: ")
         length = len(text)
        print(f"The length of the string is {length}.")
```

16. Write a Python program to count the number of characters (character frequency) in a string.

Ans:

```
text = input("Enter a string: ")
freq = {}
for char in text:
    if char in freq:
        freq[char] += 1
    else:
        freq[char] = 1
  print("Character Frequency:")
  for char, count in freq.items():
   print(f"{char}: {count}")
```

17. What are negative indexes and why are they used?

Ans: In Python, negative indexes allow you to access elements of a sequence (like a list, tuple, or string) from the end instead of the beginning.

Easy access from the end

- You don't need to calculate len(sequence) - 1 to get the last element.

Simplifies code

- Useful for operations like slicing, reversing, or checking last elements.

Supports negative slicing

- You can extract portions from the end using negative indexes.

18. Write a Python program to count occurrences of a substring in a string.

Ans: text = input("Enter a string: ")

```
 substring = input("Enter the substring to count: ")
 count = text.count(substring)
print(f"The substring '{substring}' occurs {count} times in the given string.")
```

19. Write a Python program to count the occurrences of each word in a given sentence.

Ans:

```
sentence = sentence.lower()
words = sentence.split()
word_count = {}
for word in words:
    if word in word_count:
       word_count[word] += 1
    else:
     word_count[word] = 1
print("Word Frequency:")
for word, count in word_count.items():
    print(f"{word}: {count}")
```

20. Write a Python program to get a single string from two given strings, separated by a space and swap the first two characters of each string.

Ans:

```
str1 = input("Enter first string: ")
str2 = input("Enter second string: ")
new_str1 = str2[:2] + str1[2:]
new_str2 = str1[:2] + str2[2:]
result = new_str1 + " " + new_str2
print("Result:", result)
```

21. Write a Python program to add 'in' at the end of a given string (length should be at least 3). If the given string already ends with 'ing' then add 'ly' instead if the string length of the given string is less than 3, leave it unchanged.

Ans:

```
text = input("Enter a string: ")

if len(text) >= 3:

    if text.endswith("ing"):

      text += "ly"

    else:

     text += "in"

print("Result:", text)
```

22. Write a Python function to reverses a string if its length is a multiple of 4.

Ans:

```
def reverse_if_multiple_of_4(s):

    if len(s) % 4 == 0:

       return s[::-1]  # reverse the string

    else:

       return s  # return original string

    text = input("Enter a string: ")

   result = reverse_if_multiple_of_4(text)

  print("Result:", result)
```

23. Write a Python program to get a string made of the first 2 and the last 2 chars from a given a string. If the string length is less than 2, return instead of the empty string.

Ans:
```
text = input("Enter a string: ")
 if len(text) < 2:
    result = ""
else:
  result = text[:2] + text[-2:]
print("Result:", result)
```

24. Write a Python function to insert a string in the middle of a string.

Ans:
```
def insert_middle(original, insert):
    mid = len(original) // 2  # find middle index
  return original[:mid] + insert + original[mid:]
 text1 = input("Enter the original string: ")
 text2 = input("Enter the string to insert: ")
 result = insert_middle(text1, text2)
  print("Result:", result)
```

25. What is List? How will you reverse a list?

Ans: A list is a collection of items in Python, which can store: Numbers, strings, or even other lists. Elements in a specific order. By using reverse() function, we can reverse the list.

26. How will you remove last object from a list?

Ans: By using pop() function we can remove last object from a list.

27. Suppose list1 is [2, 33, 222, 14, and 25], what is list1 [-1]?

Ans: The answer is 25.

28. Differentiate between append () and extend () methods?

Ans: In append(), it adds a single element to the end of the list. If you pass another list, it adds the whole list as one element (nested list). Where as in extend(), it adds multiple elements (from an iterable like list, tuple, or string) to the list. It unpacks the iterable and adds each element individually.

29. Write a Python function to get the largest number, smallest num and sum of all from a list.

Ans:
```
def analyze_list(numbers):
    largest = max(numbers)
    smallest = min(numbers)
    total = sum(numbers)
    return largest, smallest, total
```

30. How will you compare two lists?

Ans: By using some comparator operator we can compare.

31. Write a Python program to count the number of strings where the string length is 2 or more and the first and last character are same from a given list of strings.

Ans:
```
def count_strings(words):
    count = 0
    for word in words:
        if len(word) >= 2 and word[0] == word[-1]:
         count += 1
return count
```

32. Write a Python program to remove duplicates from a list.

Ans:

```
def remove_duplicates(my_list):
    return list(set(my_list))
```

33. Write a Python program to check a list is empty or not.

Ans:

```
def check_empty(my_list):
    if not my_list:   # empty list is considered False
            print("The list is empty")
  else:
     print("The list is not empty")
```

34. Write a Python function that takes two lists and returns true if they have at least one common member.

Ans:

```
def common_member(list1, list2):
     for item in list1:
          if item in list2:
             return True
        return False
```

35. Write a Python program to generate and print a list of first and last 5 elements where the values are square of numbers between 1 and 30.

Ans:
```
def square_list():
    squares = [x**2 for x in range(1, 31)]
    print("First 5 elements:", squares[:5])
    print("Last 5 elements:", squares[-5:])
```

36. Write a Python function that takes a list and returns a new list with unique elements of the first list.

Ans:
```
def unique_elements(my_list):
    return list(set(my_list))
```

37. Write a Python program to convert a list of characters into a string.

Ans:
```
char_list = ['P', 'y', 't', 'h', 'o', 'n']
result = ''.join(char_list)
print(result)
```

38. Write a Python program to select an item randomly from a list.

Ans:
```
import random
my_list = [10, 20, 30, 40, 50]
random_item = random.choice(my_list)
print("Randomly selected item:", random_item)
```

39. Write a Python program to find the second smallest number in a list.

Ans:
```
def second_smallest(numbers):
    unique_numbers = sorted(set(numbers))
    if len(unique_numbers) < 2:
        return None  # Not enough elements
  return unique_numbers[1]  # second smallest
```

40. Write a Python program to get unique values from a list

Ans:

```
def unique_values(my_list):
    return list(set(my_list))
```

41. Write a Python program to check whether a list contains a sub list

Ans:

```
def contains_sublist(main_list, sub_list):
     sub_len = len(sub_list)
    for i in range(len(main_list) - sub_len + 1):
        if main_list[i:i + sub_len] == sub_list:
            return True
  return False
```

42. Write a Python program to split a list into different variables.

Ans:

```
  my_list = [10, 20, 30]
 a, b, c = my_list
 print("a =", a)
print("b =", b)
print("c =", c)
```

43. What is tuple? Difference between list and tuple.

Ans: Tuple is a data collector used o store multiple data types. In tuple we use ‘()’ where as in list we use ‘[]’. The data in Tuple can not be changed but in list we can change the data.

44. Write a Python program to create a tuple with different data types.

Ans: my_tuple = (10, 3.14, "Python", True)

45. Write a Python program to unzip a list of tuples into individual lists.

Ans: tuple_list = [(1, 'a'), (2, 'b'), (3, 'c')]

```
numbers, letters = zip(*tuple_list)
numbers = list(numbers)
letters = list(letters)
print("Numbers:", numbers)
print("Letters:", letters)
```

46. Write a Python program to convert a list of tuples into a dictionary.

Ans: my_dict = dict(("a", 1), ("b", 2), ("c", 3))

```
Print(my_dict)
```

47. How will you create a dictionary using tuples in python?

Ans: By using dict() function we can change any data type into dictionary.

48. Write a Python script to sort (ascending and descending) a dictionary by value.

Ans: my_dict = {'apple': 10, 'banana': 5, 'cherry': 20, 'date': 15}

```
print("Ascending:", sorted(my_dict.items(), key=lambda x: x[1]))
print("Descending:",sorted(my_dict.items(),key=lambdax:x[1],reverse=True))
```

49. Write a Python script to concatenate following dictionaries to create a new one.

Ans:

```
dict1 = {'a': 1, 'b': 2}
dict2 = {'c': 3, 'd': 4}
dict3 = {'e': 5, 'f': 6}
new_dict = {**dict1, **dict2, **dict3}
print(new_dict)
```

50. Write a Python script to check if a given key already exists in a dictionary.

Ans:

```
my_dict = {'a': 1, 'b': 2, 'c': 3}
key = input(“Enter the key you want to check”)
if key in my_dict:
  print(f"Key '{key}' exists in the dictionary")
else:
  print(f"Key '{key}' does not exist in the dictionary")
```

51. How Do You Traverse Through a Dictionary Object in Python?

Ans:

```
my_dict = {'a': 1, 'b': 2, 'c': 3}
for key in my_dict:
  print(key, my_dict[key])
```

52. How Do You Check the Presence of a Key in A Dictionary?

Ans: By using ‘in’ operator or get() function we can check.

53. Write a Python script to print a dictionary where the keys are numbers between 1 and 15.

Ans:
```
my_dict = {x: x**2 for x in range(1, 16)}
print(my_dict)
```

54. Write a Python program to check multiple keys exists in a dictionary

Ans:
```
my_dict = {'a': 1, 'b': 2, 'c': 3}
keys = ['a', 'c', 'e']
print(all(key in my_dict for key in keys))  # False
print(any(key in my_dict for key in keys))  # True
```

55. Write a Python script to merge two Python dictionaries

Ans:
```
dict1 = {'a': 1, 'b': 2}
dict2 = {'c': 3, 'd': 4}
merged_dict = {**dict1, **dict2}
print(merged_dict)
```

56. Write a Python program to map two lists into a dictionary, Sample output: Counter ({'a': 400, 'b': 400,’d’: 400, 'c': 300}).

Ans:
```
keys = ['a', 'b', 'c', 'd']
values = [400, 400, 300, 400]
mapped_dict = dict(zip(keys, values))
print(mapped_dict)
```

57. Write a Python program to find the highest 3 values in a dictionary

Ans:
```
my_dict = {'a': 100, 'b': 200, 'c': 300, 'd': 400, 'e': 250}

        top_3_values = sorted(my_dict.values(), reverse=True)[:3]

        print("Highest 3 values:", top_3_values)
```

58. **Write a Python program to combine values in python list of dictionaries. Sample data: [{'item': 'item1', 'amount': 400}, {'item': 'item2', 'amount': 300}, o {'item': 'item1', 'amount': 750}] Expected Output: • Counter ({'item1': 1150, 'item2': 300})**

Ans:
```
data = [

    {'item': 'item1', 'amount': 400},

    {'item': 'item2', 'amount': 300},

    {'item': 'item1', 'amount': 750}]

combined = {}

for d in data:

    if d['item'] in combined:

        combined[d['item']] += d['amount']

    else:

        combined[d['item']] = d['amount']

print(combined)
```

59. Write a Python program to create a dictionary from a string.

Ans:
```
from collections import Counter

text = "hello world"

letter_count = Counter(text.replace(" ", ""))  # remove spaces

print(letter_count)
```

60. Sample string: 'w3resource' Expected output: • {'3': 1,’s’: 1, 'r': 2, 'u': 1, 'w': 1, 'c': 1, 'e': 2, 'o': 1}

Ans:
```
text = "w3resource"

char_count = {}

for char in text:

    char_count[char] = char_count.get(char, 0) + 1

print(char_count)
```

61. Write a Python function to calculate the factorial of a number (a nonnegative integer)

Ans:
```
def factorial(n):

 if n < 0:

     return "Factorial not defined for negative numbers"

 elif n == 0 or n == 1:

     return 1

 else:               result = 1

for i in range(2, n + 1):

    result *= i

return result
```

62. Write a Python function to check whether a number is in a given range

Ans:
```
def check_in_range(num, start, end):
    if num in range(start, end+1):
        return True
    else:
        return False
```

63. Write a Python function to check whether a number is perfect or not.

Ans:
```
def is_perfect(num):
    return num == sum(i for i in range(1, num) if num % i == 0)
```

64. Write a Python function that checks whether a passed string is palindrome or not

Ans:
```
def is_palindrome(s):
    s = s.lower()  # ignore case
    return s == s[::-1]
```

65. How Many Basic Types of Functions Are Available in Python?

Ans: There are two types of functions. One is Built-in Function, another is User-defined Function.

66. How can you pick a random item from a list or tuple?

Ans: By using random.choice()

67. How can you pick a random item from a range?

Ans: By using random.randrange(start, stop)

68. How can you get a random number in python?

Ans: There are several functions such as random.random(), randint(a, b), uniform(a, b)

69. How will you set the starting value in generating random numbers?

Ans: To set the starting value in generating random numbers, you use random.seed() from Python’s random module.

70. How will you randomize the items of a list in place?

Ans: By using random.shuffle()

71. What is File function in python? What are keywords to create and write file.

Ans: File functions in python means the function to work on any file. Weather to create or write or read or append. To create file, use open(), to write use ‘w’.

72. Write a Python program to read an entire text file.

Ans:

```
f = open("example.txt", "r")
print(f.read())
f.close()
```

73. Write a Python program to append text to a file and display the text.

Ans:

```
f = open("example.txt", "a")
 f.write("This is new text.\n")
f.close()
f = open("example.txt", "r")
print(f.read())
 f.close()
```

74. Write a Python program to read first n lines of a file.

Ans: n = 3

```
f = open("example.txt", "r")

for i in range(n):

    print(f.readline(), end="")

f.close()
```

75. Write a Python program to read last n lines of a file.

Ans: n = 3

```
         f = open("example.txt", "r")

        lines = f.readlines()

       for line in lines[-n:]:

          print(line, end="")

       f.close()
```

76. Write a Python program to read a file line by line and store it into a list

Ans: f = open("example.txt", "r")

```
         lines = f.readlines()  # Reads all lines into a list

         f.close()

        print(lines)
```

77. Write a Python program to read a file line by line store it into a variable.

Ans: f = open("example.txt", "r")

```
       content = f.read()  # Reads entire file into a variable

      f.close()   print(content)
```

78. Write a python program to find the longest words.

Ans:

```
text = "Python is an amazing programming language"
words = text.split()
longest = max(words, key=len)
print(longest)
```

79. Write a Python program to count the number of lines in a text file.

Ans:

```
f = open("example.txt", "r")
lines = f.readlines()
print("Number of lines:", len(lines))
f.close()
```

80. Write a Python program to count the frequency of words in a file.

Ans:

```
     f = open("example.txt", "r")
     words = f.read().split()
     f.close()
    freq = {}
  for word in words:
     if word in freq:
        freq[word] += 1
    else:
       freq[word] = 1
print(freq)
```

81. Write a Python program to write a list to a file.

Ans:
```
my_list = ["apple", "banana", "cherry"]
     f = open("example.txt", "w")
    for item in my_list:
       f.write(item + "\n")  # Write each item on a new line
    f.close()
```

82. Write a Python program to copy the contents of a file to another file.

Ans:
```
f1 = open("source.txt", "r")
      f2 = open("destination.txt", "w")
     f2.write(f1.read())
     f1.close()
    f2.close()
   print("File copied successfully!")
```

83. Explain Exception handling? What is an Error in Python?

Ans: Exception handling in python means to control the errors. Error means that stops the code.

84. How many except statements can a try-except block have? Name Some built-in exception classes:

Ans: A try block can have multiple except statements to handle different types of exceptions. There are VaueError, ZeroDivisionError etc.

85. When will the else part of try-except-else be executed?

Ans: In Python, the else part of a try-except-else block is executed only if no exception occurs in the try block.

86. Can one block of except statements handle multiple exception?

Ans: Yes, in Python, one except block can handle multiple exceptions by specifying them as a tuple.

87. When is the finally block executed?

Ans: In Python, the finally block is executed no matter what happens—whether an exception occurs or not.

88. What happens when “1‟== 1 is executed?

Ans: Python will compare a string '1' with an integer 1.

89. How Do You Handle Exceptions with Try/Except/Finally in Python? Explain with coding snippets.

Ans:

```
try:
    x = int(input("Enter a number: "))
    print(10 / x)
except:
    print("An error occurred!")
finally:
    print("This always runs.")
```

90. Write python program that user to enter only odd numbers, else will raise an exception.

Ans:

```
try:
    num = int(input("Enter an odd number: "))
    if num % 2 == 0:
        raise Exception("You entered an even number!")
    print("You entered:", num)

except Exception as e:

    print(e)
```