**Assignment -1**

Q1. Why do we call Python as a general purpose and high-level programming language?

Ans : Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

**Python is a very high-level programming language because its syntax so closely resembles the English language. Higher-level means it’s more readable to humans and less readable to computers. Likewise, Lower-level means less readable for humans and more readable for computers.**

**Q2. Why is Python called a dynamically typed language?**

**Ans :** But Python is a **dynamically typed** language. It doesn’t know about the type of the variable until the code is run. So declaration is of no use. What it does is, It stores that value at some memory location and then binds that variable name to that memory container. And makes the contents of the container accessible through that variable name. So the data type does not matter

Q3: List some pros and cons of Python programming language?

Ans : Pros and cron of python :

. Python is easy to read and learn.

. Python is enhances productivity

. Python is open sources**.**

. Python has a vast collection of libraries

. Python has speed limitations

.Consume lots of memory space

Q4. In what all domains can we use Python?

Ans: Machine learning / Artificial intelligence, Data analytics and data visualization, Web development,

Game development, Mobile app development.

Q5. What are variable and how can we declare them?

Ans : A variable declaration always contains two components: the type of the variable and its name. Also, the location of the variable declaration, that is, where the declaration appears in relation to other code elements, determines the scope of the variable.

Example : x=5 ,y=3 and z=’RAM’

Q6.How can we take an input from the user in Python?

Ans: In python for taking input Basically declare a variable and print the variable also use input() statement.

inp = input('STATEMENT')

Q7. What is the default datatype of the value that has been taken as an input using input() function?

Ans : Intiger,float,string,Bolien etc.

Q8. What is type casting?

Ans: Type Casting is the method to convert the variable data type into a certain data type in order to the operation required to be performed by users.

Q9. Can we take more than one input from the user using single input() function? If yes, how? If no, why?

Ans: Yes, using split() function.

input().split(separator, maxsplit)

Q10. What are keywords?

Ans : Value ,Operator ,Control Flow ,Iteration ,Structure ,Import

Q11: Can we use keywords as a variable? Support your answer with reason.

Ans : We cannot use a keyword as a variable name, function name, or any other identifier. They are used to define the syntax and structure of the Python language.

Q12. What is indentation? What's the use of indentaion in Python?

Ans : Indentation refers to the spaces at the beginning of a code line. Where in other programming languages the indentation in code is for readability only, the indentation in Python is very important. Python uses indentation to indicate a block of code.

Q13: How can we throw some output in Python?

Ans : In python Basically use print() statement for throw output.

Q14. What are operators in Python?

Ans : Numerical operators in Python, Comparison Operators, Assignment operators,Logical operator.

Q15. What is difference between / and // operators?

Ans “/” is division operator and “//” is floor division operator.

Q16. Write a code that gives following as an output?

iNeuroniNeuroniNeuroniNeuron

Ans : a='iNeuron'

b=a\*4

print(b)

Q17. Write a code to take a number as an input from the user and check if the number is odd or even.

Ans: num= int(input())

if num%2==0:

print("even")

else:

print("odd")

Q18. What are boolean operator?

Ans: Boolean operators are words that connect search terms (keywords) to create a logical phrase that a database can understand. They allow you to create a complex search that could include multiple concepts and alternative keywords.

Q19. What will the output of the following?

1 or 0

0 and 0

True and False and True

1 or 0 or 0

Ans : 1

0

False

1

Q20. What are conditional statements in Python?

Ans: Equals,Not Equals,Less than,Less than or equal to,Greater than ,Greater than or equal to

Q21. What is use of 'if', 'elif' and 'else' keywords?

Ans: if… elif…else are conditional statements that **provide you with the decision making that is required when you want to execute code based on a particular condition**.

Q22: Write a code to take the age of person as an input and if age >= 18 display "I can vote". If age is < 18 display "I can't vote"?

Ans : age=int(input(" age "))

if age>=18:

   print('I can Vote')

else:

  print("I can't vote")

Q23: Write a code that displays the sum of all the even numbers from the given list.

numbers = [12, 75, 150, 180, 145, 525, 50]

Ans: numbers = [12, 75, 150, 180, 145, 525, 50]

even\_sum=sum(x for x in numbers if x%2==0)

print(even\_sum)

Q24: Write a code to take 3 numbers as an input from the user and display the greatest no as output.

Ans: a = int(input("Please enter the first number: "))

b = int(input("Please enter the second number: "))

c = int(input("Please enter the third number: "))

if a > b and a > c:

    print(a, "is the greatest number.")

elif b > a and b > c:

    print(b, "is the greatest number.")

else:

    print(c, "is the greatest number.")

Q25: Q25. Write a program to display only those numbers from a list that satisfy the following conditions

* The number must be divisible by five
* If the number is greater than 150, then skip it and move to the next number
* If the number is greater than 500, then stop the loop

Ans: numbers = [12, 75, 150, 180, 145, 525, 50, 550, 1000]

for number in numbers:

    if number % 5 == 0:

        if number > 150 and number <= 500:

            print(number)

        elif number > 500:

            break

Q26. What is a string? How can we declare string in Python? Ans:A string is a sequence of characters in programming. It is a data  type that can store a series of letters, numbers, and special characters. Strings are often used to represent words, phrases, sentences, or any other type of text.

In Python, a string can be declared by enclosing a set of characters in single quotes ('...') or double quotes ("...").

Q27. How can we access the string using its index?

Ans: In Python, strings are indexed, which means that each character within the string has a corresponding numerical index, starting from 0.

You can access individual characters in a string using square brackets [] and the index of the character you want to access

string = 'Hello World'

# Indexing starts from 0

print(string[0]) # Output: 'H'

print(string[1]) # Output: 'e'

print(string[2]) # Output: 'l'

Q28. Write a code to get the desired output of the following?

string = "Big Data iNeuron"

desired\_output = "iNeuron"

Ans : string = "Big Data iNeuron"

desired\_output = string[9:]

print(desired\_output) # Output: 'iNeuron'

29.Write a code to get the desired output of the following

string = "Big Data iNeuron"

desired\_output = "norueNi"

Ans: string = "Big Data iNeuron"

# Using string slicing

desired\_output = string[9:][::-1]

print(desired\_output) # Output: 'norueNi'

30. Resverse the string given in the above question.

Ans: string = "Big Data iNeuron"

desired\_output = string[::-1]

print(desired\_output)

Q31. How can you delete entire string at once?

Ans: use del keyword.

Q32. What is escape sequence?

Ans : An escape sequence is a sequence of characters that, when used inside a character or string, does not represent itself but is converted into another character or series of characters.

Q33. How can you print the below string?

'iNeuron's Big Data Course'

Ans:s='iNeuron's Big Data Course'

Print(s)

Q34. What is a list in Python?

Ans: A list in Python is used to store the sequence of various types of data. A list can be defined as a collection of values or items of different types. Python lists are mutable type which implies that we may modify its element after it has been formed. The items in the list are separated with the comma (,) and enclosed with the square brackets [].

Q35. How can you create a list in Python?

Ans: names = ["Jane", "John", "Jade", "Joe"]

Q36 : Q36. How can we access the elements in a list?

Ans : usning index .like names[0]

Q37. Write a code to access the word "iNeuron" from the given list.

Ans : lst = [1,2,3,"Hi",[45,54, "iNeuron"], "Big Data"]

Print( lst[4][2])

Q38. Take a list as an input from the user and find the length of the list.

Ans: dl= print(“enter the number :”).split(“,”)

Print(len(dl))

Q39. Add the word "Big" in the 3rd index of the given list.

Ans: lst = ["Welcome", "to", "Data", "course"]

lst.insert(2,’big’)

print(lst)

Q40. What is a tuple? How is it different from list?

Ans: Tuples are immutable, while lists are mutable.

* Tuples use less memory and are faster to access than lists.
* Tuples are typically used to store related but different pieces of data, while lists are used to store similar pieces of data.
* Tuples can be used as keys in a dictionary, whereas lists cannot.

Q41. How can you create a tuple in Python?

Ans: F=(“ravi”,’syam’,’ram’)

Q42. Create a tuple and try to add your name in the tuple. Are you able to do it? Support your answer with reason.

Ans: showing error .because tuple are immuatable.

Q43. Can two tuple be appended. If yes, write a code for it. If not, why?

Ans: Yes,

tuple1 = (1, 2, 3)

tuple2 = (4, 5, 6)

result = tuple1 + tuple2

print(result)

Q44. Take a tuple as an input and print the count of elements in it.

Ans: my\_tuple = tuple(input("Enter elements of tuple, separated by commas: ").split(",")) print(len(my\_tuple))

Q45. What are sets in Python?

Ans: n Python, a set is a collection of unique elements. Sets are similar to lists and tuples in that they can store multiple items, but unlike lists and tuples, sets do not maintain any order among the elements, and elements in a set are unique, meaning no element can be repeated. Sets are also mutable, which means that elements can be added, removed, or modified after the set has been created.

Sets are created using the set() constructor or curly braces {} with elements .

Q45 How can you create a set?

Ans: thisset = {"apple", "banana", "cherry"}  
 print(thisset)

Q47. Create a set and add "iNeuron" in your set.

Ans: my\_set = {1, 2, 3}

my\_set = my\_set.union(["iNeuron"])

print(my\_set)

Q48. Try to add multiple values using add() function.

Ans: my\_set = {1, 2, 3}

my\_set.add("iNeuron")

my\_set.add("Data Science")

my\_set.add("Python")

print(my\_set)

Q49. How is update() different from add()?

* Ans: Use add() function to add a single element. Whereas use update() function to add multiple elements to the set.
* add() is faster than update().
* add () accepts immutable parameters only. Whereas accepts iterable sequences.
* add() accepts a single parameter, whereas update() can accept multiple sequences.

Q50. What is clear() in sets?

Ans: The clear() method removes all elements in a set.

Q51. What is frozen set?

Ans: Python frozenset()Methodcreates an immutable Set object from an iterable. It is a built-in Python function. As it is a set object therefore we cannot have duplicate values in the frozenset.

Q52. How is frozen set different from set?

Ans: Frozenset is similar to set in Python, except that frozensets are immutable, which implies that once generated, elements from the frozenset cannot be added or removed. This function accepts any iterable object as input and transforms it into an immutable object.

Q53. What is union() in sets? Explain via code.

Ans: Union of two given sets is the set which contains all the elements of both the sets. The union of two given sets A and B is a set which consists of all the elements of A and all the elements of B such that no element is repeated.

A = {2, 4, 5, 6}

B = {4, 6, 7, 8}

print("A U B:", A.union(B))

output: A U B: {2, 4, 5, 6, 7, 8}

Q54. What is intersection() in sets? Explain via code.

Ans: The intersection of two given sets is the largest set, which contains all the elements that are common to both sets. The intersection of two given sets A and B is a set which consists of all the elements which are common to both A and B.

s1 = {1, 2, 3}

s2 = {2, 3}

print(s1.intersection(s2))

Q55. What is dictionary ibn Python?

Ans: Dictionaries are Python’s implementation of a data structure that is more generally known as an associative array. A dictionary consists of a collection of key-value pairs. Each key-value pair maps the key to its associated value.

You can define a dictionary by enclosing a comma-separated list of key-value pairs in curly braces ({}). A colon (:) separates each key from its associated value:

Q56. How is dictionary different from all other data structures.

Ans: A dictionary is a data structure that stores elements as key-value pairs, where each key is unique and can be used to access its corresponding value. Unlike other data structures such as lists, arrays, and sets, dictionaries are unordered and do not maintain any index-based relationship between elements. Additionally, dictionaries allow constant-time lookup and update operations based on keys, making them efficient for searching and mapping values in large collections of data.

Q57. How can we delare a dictionary in Python?

Ans: my\_dict = {'key1': 'value1', 'key2': 'value2', 'key3': 'value3'

Q58. What will the output of the following?

var = {}

print(type(var))

Ans: dict

Q59. How can we add an element in a dictionary?

Ans: To add an element in a dictionary, you can use square brackets **[]** with the key to assign a value. For example:

d = {}

d["key"] = "value"

Q60. Create a dictionary and access all the values in that dictionary.

Ans: d = {"key1": "value1", "key2": "value2", "key3": "value3"}

for value in d.values():

print(value)

Q61. Create a nested dictionary and access all the element in the inner dictionary.

Ans: d = {"key1": {"inner\_key1": "inner\_value1", "inner\_key2": "inner\_value2"},

"key2": {"inner\_key3": "inner\_value3", "inner\_key4": "inner\_value4"}}

for key, value in d.items():

print(key, value)

Q62. What is the use of get() function?

Ans: The get() function is used to access the value of a key in a dictionary. If the key is not present in the dictionary, it returns None or a default value specified by the user.

Q63. What is the use of items() function?

Ans: The **items()** function returns a view of the dictionary's key-value pairs as a list of tuples

Q64. What is the use of pop() function?

Ans: The pop() function is used to remove and return the value of a specified key in a dictionary. If the key is not present in the dictionary, it raises a KeyError or returns a default value specified by the user.

Q65. What is the use of popitems() function?

Ans: The popitem() function is used to remove and return an arbitrary (key, value) pair from the dictionary.

Q66. What is the use of keys() function?

Ans: The **keys()** function returns a view of the dictionary's keys.

Q67. What is the use of values() function?

Ans: The **values()** function returns a view of the dictionary's values.

Q68. What are loops in Python?

Ans: for loop,while loo[s

Q69. How many type of loop are there in Python?

Ans: There are two types of loops in Python: for loop and while loop.

Q70. What is the difference between for and while loops?

Ans: A for loop is used to iterate over a sequence (such as a list, tuple, string, or range) and execute a block of code for each item in the sequence. It is used when the number of iterations is known in advance.

A while loop is used to repeatedly execute a block of code as long as a given condition is true. It is used when the number of iterations is not known in advance.

In simple words, a for loop is used when we know the number of iterations, whereas a while loop is used when we don't know the number of iterations.

Q71. What is the use of continue statement?

Ans : The "continue" statement is used in a loop to skip the current iteration and move on to the next iteration. It is used to bypass the current iteration of the loop and continue to the next iteration, without executing the code after the continue statement in the current iteration. For example, if you have a loop to traverse a list, and you want to skip certain elements, you can use the continue statement to move on to the next element.

Q72. What is the use of break statement?

Ans: The "break" statement is used to stop the execution of a loop, and immediately jump out of the loop to the next statement after the loop. It is used inside the loop body to exit the loop when a certain condition is met. For example, when searching for a specific item in a list, you can use the "break" statement to exit the loop when the item is found, instead of looping through all the items.

Q73. What is the use of pass statement?

Ans: The pass statement is a null operation, a placeholder used when a statement is required but no code needs to be executed. It acts as a placeholder for future code and can be used to create stubs for functions, classes, and other constructs. In other words, pass can be used to create a non-executable code block, which is useful in cases where the code is not yet implemented but you want to keep the syntax and structure of the code intact.

Q74. What is the use of range() function?

Ans: The range() function in Python is used to generate a sequence of numbers. It takes two or three parameters: start, stop, and step. The start parameter is the first number in the sequence, the stop parameter is the first number that is not in the sequence, and the step parameter is the difference between each number in the sequence. The range() function can be used in for loops to iterate over the sequence of numbers generated by the function.

Q75. How can you loop over a dictionary?

Ans: my\_dict = {'key1': 'value1', 'key2': 'value2', 'key3': 'value3'}

for key, value in my\_dict.items():

print(key, value)

Q76. Write a Python program to find the factorial of a given number.

Ans: def factorial(n):

if n == 0:

return 1

else:

result = 1

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

result = result \* i

return result

num = int(input("Enter a number: "))

print("The factorial of", num, "is", factorial(num))

Q77. Write a Python program to calculate the simple interest. Formula to calculate simple interest is SI = (PRT)/100

Ans: def simple\_interest(principal, rate, time):

si = (principal \* rate \* time) / 100

return si

p = float(input("Enter the principal amount: "))

r = float(input("Enter the rate of interest: "))

t = float(input("Enter the time period (in years): "))

result = simple\_interest(p, r, t)

print("Simple Interest:", result)

Q78. Write a Python program to calculate the compound interest. Formula of compound interest is A = P(1+ R/100)^t.

Ans: def compound\_interest(principal, rate, time):

return principal \* (1 + rate/100) \*\* time

principal = float(input("Enter the principal amount: "))

rate = float(input("Enter the interest rate: "))

time = float(input("Enter the number of years: "))

result = compound\_interest(principal, rate, time)

print("The compound interest is", result)

Q79. Write a Python program to check if a number is prime or not.

Ans: def is\_prime(n):

if n <= 1:

return False

for i in range(2, n):

if n % i == 0:

return False

return True

number = int(input("Enter a number: "))

if is\_prime(number):

print(f"{number} is a prime number")

else:

print(f"{number} is not a prime number")

Q80. Write a Python program to check Armstrong Number.

Ans: def is\_armstrong\_number(number):

# Calculate the number of digits in the number

num\_of\_digits = len(str(number))

# Initialize the sum

sum = 0

# Find the sum of the digits raised to the power of the number of digits

temp = number

while temp > 0:

digit = temp % 10

sum += digit \*\* num\_of\_digits

temp //= 10

# Check if the sum is equal to the original number

if number == sum:

return True

else:

return False

# Get input from the user

number = int(input("Enter a number: "))

# Check if the number is an Armstrong number

if is\_armstrong\_number(number):

print(number, "is an Armstrong number.")

else:

print(number, "is not an Armstrong number.")

Q81. Write a Python program to find the n-th Fibonacci Number.

Ans: def fib(n):

a = 0

b = 1

for i in range(n):

a, b = b, a + b

return a

# Test the function with different values of n

n = int(input("Enter the value of n: "))

print("The", n, "th Fibonacci number is", fib(n))

Q82. Write a Python program to interchange the first and last element in a list.

Ans: def interchange\_first\_last(input\_list):

if len(input\_list) == 0:

return []

else:

first = input\_list[0]

last = input\_list[-1]

input\_list[0] = last

input\_list[-1] = first

return input\_list

input\_list = [1, 2, 3, 4, 5]

print("Original List: ", input\_list)

print("List after interchanging first and last elements: ", interchange\_first\_last(input\_list))

Q83. Write a Python program to swap two elements in a list.

Ans: def swap\_elements(lst, index1, index2):

lst[index1], lst[index2] = lst[index2], lst[index1]

return lst

# sample list

my\_list = [1, 2, 3, 4, 5]

# swapping elements

my\_list = swap\_elements(my\_list, 0, 4)

# printing the list

print("List after swapping:", my\_list)

Q84. Write a Python program to find N largest element from a list.

Ans: def N\_largest\_elements(lst, N):

sorted\_list = sorted(lst, reverse=True)

return sorted\_list[:N]

# Example usage

lst = [1, 7, 3, 9, 5]

print(N\_largest\_elements(lst, 3)) # [9, 7, 5]

Q85. Write a Python program to find cumulative sum of a list.

Ans: def cumulative\_sum(numbers):

cumulative\_sum = []

current\_sum = 0

for num in numbers:

current\_sum += num

cumulative\_sum.append(current\_sum)

return cumulative\_sum

numbers = [1, 2, 3, 4, 5]

print(cumulative\_sum(numbers))

Q86. Write a Python program to check if a string is palindrome or not.

Ans: def is\_palindrome(s):

return s == s[::-1]

string = input("Enter a string: ")

if is\_palindrome(string):

print("The string is a palindrome.")

else:

print("The string is not a palindrome.")

Q87. Write a Python program to remove i'th element from a string.

Ans: def remove\_ith\_char(string, i):

return string[:i] + string[i+1:]

string = input("Enter a string: ")

i = int(input("Enter the index of the character to be removed: "))

print("Modified string:", remove\_ith\_char(string, i))

Q88. Write a Python program to check if a substring is present in a given string.

Ans: def check\_substring(string, substring):

if substring in string:

return True

else:

return False

string = input("Enter the main string: ")

substring = input("Enter the substring to search: ")

result = check\_substring(string, substring)

if result:

print("Substring is present in the given string.")

else:

print("Substring is not present in the given string.")

Q89. Write a Python program to find words which are greater than given length k.

Ans: def find\_words\_greater\_than\_k(string, k):

words = string.split()

greater\_words = [word for word in words if len(word) > k]

return greater\_words

string = "This is a sample string to find words greater than a given length"

k = 6

print("Words greater than", k, "in length:", find\_words\_greater\_than\_k(string, k))

Q90. Write a Python program to extract unquire dictionary values.

Ans: def extract\_unique\_values(dictionary):

return set(dictionary.values())

# Sample dictionary

my\_dict = {'a': 1, 'b': 2, 'c': 3, 'd': 1}

# Extract unique values

unique\_values = extract\_unique\_values(my\_dict)

# Print the result

print(unique\_values)

Q91. Write a Python program to merge two dictionary.

Ans: def merge\_dicts(dict1, dict2):

return {\*\*dict1, \*\*dict2}

# Sample dictionaries

dict1 = {'a': 1, 'b': 2}

dict2 = {'c': 3, 'd': 4}

# Merge dictionaries

merged\_dict = merge\_dicts(dict1, dict2)

# Print the result

print(merged\_dict)

Q92. Write a Python program to convert a list of tuples into dictionary.

Input : [('Sachin', 10), ('MSD', 7), ('Kohli', 18), ('Rohit', 45)]

Output : {'Sachin': 10, 'MSD': 7, 'Kohli': 18, 'Rohit': 45}

Ans: def list\_to\_dict(list\_of\_tuples):

return {key: value for key, value in list\_of\_tuples}

# Sample list of tuples

list\_of\_tuples = [('Sachin', 10), ('MSD', 7), ('Kohli', 18), ('Rohit', 45)]

# Convert list to dictionary

dict\_from\_list = list\_to\_dict(list\_of\_tuples)

# Print the result

print(dict\_from\_list)

Q93. Write a Python program to create a list of tuples from given list having number and its cube in each tuple.

Input: list = [9, 5, 6]

Output: [(9, 729), (5, 125), (6, 216)]

Ans: def number\_and\_cube(numbers):

return [(num, num\*\*3) for num in numbers]

# Sample list

numbers = [9, 5, 6]

# Create list of tuples

list\_of\_tuples = number\_and\_cube(numbers)

# Print the result

print(list\_of\_tuples)

Q94. Write a Python program to get all combinations of 2 tuples.

Input : test\_tuple1 = (7, 2), test\_tuple2 = (7, 8)

Output : [(7, 7), (7, 8), (2, 7), (2, 8), (7, 7), (7, 2), (8, 7), (8, 2)]

Ans: def tuple\_combinations(tuple1, tuple2):

return [(x, y) for x in tuple1 for y in tuple2] + [(y, x) for x in tuple1 for y in tuple2]

# Sample tuples

tuple1 = (7, 2)

tuple2 = (7, 8)

# Get all combinations of tuples

combinations = tuple\_combinations(tuple1, tuple2)

# Print the result

print(combinations)

Q95. Write a Python program to sort a list of tuples by second item.

Input : [('for', 24), ('Geeks', 8), ('Geeks', 30)]

Output : [('Geeks', 8), ('for', 24), ('Geeks', 30)]

Ans: def sort\_by\_second\_item(list\_of\_tuples):

return sorted(list\_of\_tuples, key=lambda x: x[1])

# Sample list of tuples

list\_of\_tuples = [('for', 24), ('Geeks', 8), ('Geeks', 30)]

# Sort the list of tuples

sorted\_list = sort\_by\_second\_item(list\_of\_tuples)

# Print the result

print(sorted\_list)

Q96. Write a python program to print below pattern.

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

Ans: for i in range(1, 6):

for j in range(i):

print("\*", end=" ")

print("")

Q97. Write a python program to print below pattern.

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

Ans: for i in range(1, 6):

for j in range(6 - i):

print(" ", end="")

for k in range(i):

print("\*", end="")

print("")

Q98. Write a python program to print below pattern.

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

Ans: for i in range(1, 6):

for j in range(6 - i):

print(" ", end="")

for k in range(i):

print("\*", end=" ")

print("")

Q99. Write a python program to print below pattern.

1

1 2

1 2 3

1 2 3 4

1 2 3 4 5

Ans: for i in range(1, 6):

for j in range(1, i + 1):

print(j, end=" ")

print("")

Q100. Write a python program to print below pattern.

A

B B

C C C

D D D D

E E E E E

Ans:

for i in range(65, 70):

for j in range(i - 64):

print(chr(i), end=" ")

print("")