**Python Assignment**

# **1) What are the types of Applications?**

**Applications can be categorized into the following types:  
- Web Applications  
- Desktop Applications  
- Mobile Applications  
- Console Applications  
- Enterprise Applications  
- Game Applications**

# **2) What is programming?**

**Programming is the process of designing and writing instructions (code) that a computer can execute to perform specific tasks or solve problems.**

# **3) What is Python?**

**Python is a high-level, interpreted, object-oriented programming language. It is widely used for web development, data science, artificial intelligence, automation, and many other fields due to its simplicity and readability.**

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

**Program:  
num = int(input('Enter a number: '))  
if num > 0:  
 print('Positive number')  
elif num < 0:  
 print('Negative number')  
else:  
 print('Zero')**

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

**Program:  
def factorial(n):  
 return 1 if n == 0 else n \* factorial(n-1)  
num = int(input('Enter a number: '))  
print('Factorial:', factorial(num))**

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

**Program:  
def fibonacci(n):  
 a, b = 0, 1  
 for \_ in range(n):  
 print(a, end=' ')  
 a, b = b, a+b  
n = int(input('Enter range: '))  
fibonacci(n)**

# **7) How memory is managed in Python?**

**Python memory management is handled by the Python Memory Manager. It involves:  
- Private heap space: All objects and data structures are stored here.  
- Automatic garbage collection: Unused memory is automatically freed.  
- Reference counting and cyclic garbage collector are used to manage memory.**

# **8) What is the purpose of continue statement in Python?**

**The 'continue' statement in Python is used to skip the current iteration of a loop and move to the next iteration. It is useful when you want to skip some part of code under certain conditions without terminating the loop.**

# **9) Write python program that swap two numbers with temp variable and without temp variable.**

**With temp variable:  
x, y = 5, 10  
temp = x  
x = y  
y = temp  
print(x, y)  
  
Without temp variable:  
x, y = 5, 10  
x, y = y, x  
print(x, y)**

# **10) Write a Python program to find whether a given number is even or odd.**

**Program:  
num = int(input('Enter a number: '))  
if num % 2 == 0:  
 print('Even number')  
else:  
 print('Odd number')**

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

**Program:  
ch = input('Enter a letter: ').lower()  
if ch in 'aeiou':  
 print('Vowel')  
else:  
 print('Not a vowel')**

# **12) Write a Python program to sum of three given integers. However, if two values are e ual sum will be zero.**

**Program:  
def custom\_sum(x, y, z):  
 if x == y or y == z or x == z:  
 return 0  
 return x + y + z  
print(custom\_sum(2, 3, 4))**

# **13) Write a Python program that will return true if the two given integer values are e ual or their sum or difference is 5.**

**Program:  
def check\_values(x, y):  
 return x == y or abs(x-y) == 5 or (x+y) == 5  
print(check\_values(2, 3))**

# **14) Write a Python program to sum of the first n positive integers.**

**Program:  
n = int(input('Enter n: '))  
sum\_n = n \* (n + 1) // 2  
print('Sum:', sum\_n)**

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

**Program:  
s = input('Enter a string: ')  
print('Length of string:', len(s))**

# **16) Write a Python program to count the number of characters (character fre uency) in a string.**

**Program:  
from collections import Counter  
s = input('Enter a string: ')  
print(Counter(s))**

# **17) What are negative indexes and why are they used?**

**Negative indexes in Python allow you to access elements from the end of a se uence. For example, list[-1] refers to the last element, list[-2] refers to the second last element. They are useful when you want to access elements from the end without knowing the exact length.**

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

**Program:  
s = 'hello world, hello python'  
print(s.count('hello'))**

# **19) Write a Python program to count the occurrences of each word in a string.**

**Program:  
from collections import Counter  
s = 'this is a test this is python'  
words = s.split()  
print(Counter(words))**

# **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.**

**Program:  
def swap\_strings(a, b):  
 return b[:2] + a[2:] + ' ' + a[:2] + b[2:]  
print(swap\_strings('abc', 'xyz'))**

# **21) Write a Python program to add 'in' at the end of a given string.**

**Program:  
def add\_string(s):  
 if len(s) < 3:  
 return s  
 if s.endswith('ing'):  
 return s + 'ly'  
 return s + 'ing'  
print(add\_string('play'))**

# **22) Write a Python function to reverse a string if its length is a multiple of 4.**

**Program:  
def reverse\_if\_multiple\_of\_4(s):  
 return s[::-1] if len(s) % 4 == 0 else s  
print(reverse\_if\_multiple\_of\_4('abcd'))**

# **23) Write a Python program to get a string made of the first 2 and the last 2 chars from a given string.**

**Program:  
def first\_last\_chars(s):  
 return '' if len(s) < 2 else s[:2] + s[-2:]  
print(first\_last\_chars('python'))**

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

**Program:  
def insert\_middle(s, word):  
 mid = len(s) // 2  
 return s[:mid] + word + s[mid:]  
print(insert\_middle('python', 'JAVA'))**

# **25) What is List? How will you reverse a list?**

**A list in Python is a collection of ordered, mutable items. Lists can store elements of different data types.  
To reverse a list:  
list1 = [1,2,3,4]  
list1.reverse()  
or  
list1[::-1]**

# **26) How will you remove last object from a list?**

**We can use pop():  
list1 = [1,2,3]  
list1.pop()**

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

**list1[-1] gives the last element of the list. So, output = 25.**

# **28) Differentiate between append() and extend() methods?**

**- append(): Adds its argument as a single element to the end of the list.  
- extend(): Iterates over its argument adding each element to the list.**

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

**Program:  
def list\_operations(lst):  
 return max(lst), min(lst), sum(lst)  
print(list\_operations([1,2,3,4,5]))**

# **30) How will you compare two lists?**

**Two lists can be compared using '==' operator to check if elements are e ual and in the same order:  
list1 == list2**

# **31) Write a Python program to count strings where string length >=2 and first and last character are same.**

**Program:  
def match\_words(words):  
 count = 0  
 for w in words:  
 if len(w) >= 2 and w[0] == w[-1]:  
 count += 1  
 return count  
print(match\_words(['abc','xyz','aba','1221']))**

# **32) Write a Python program to remove duplicates from a list.**

**Program:  
list1 = [1,2,2,3,4,4]  
list1 = list(set(list1))  
print(list1)**

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

**list1 = []  
if not list1:  
 print('Empty')  
else:  
 print('Not Empty')**

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

**def common\_member(a, b):  
 return any(i in a for i in b)  
print(common\_member([1,2,3],[4,5,3]))**

# **35) Write a Python program to generate list of s uares of numbers between 1 and 30.**

**s uares = [x\*\*2 for x in range(1,31)]  
print(s uares[:5])  
print(s uares[-5:])**

# **36) Write a Python function that takes a list and returns a new list with uni ue elements.**

**def uni ue\_list(lst):  
 return list(set(lst))  
print(uni ue\_list([1,2,2,3,4]))**

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

**s = ['h','e','l','l','o']  
print(''.join(s))**

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

**import random  
items = [1,2,3,4,5]  
print(random.choice(items))**

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

**lst = [1, 2, 3, 4]  
lst.sort()  
print(lst[1])**

# **40) Write a Python program to get uni ue values from a list.**

**lst = [1,2,2,3,4,4]  
print(list(set(lst)))**

# **41) Write a Python program to check whether a list contains a sublist.**

**def is\_sublist(lst, sub):  
 return any(lst[i:i+len(sub)] == sub for i in range(len(lst)-len(sub)+1))  
print(is\_sublist([1,2,3,4],[2,3]))**

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

**lst = [1,2,3]  
a, b, c = lst  
print(a, b, c)**

# **43) What is tuple? Difference between list and tuple.**

**Tuple: An ordered, immutable collection of items.  
Difference:  
- Lists are mutable, tuples are immutable.  
- Lists use [], tuples use ().**

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

**t = (1, 'hello', 3.14, True)  
print(t)**

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

**lst = [(1,2),(3,4),(5,6)]  
a, b = zip(\*lst)  
print(list(a), list(b))**

## **47) How will you create a dictionary using tuples in python?**

**tuples = (('name', 'Nikunj'), ('age', 25), ('course', 'Data Analysis'))  
dictionary = dict(tuples)  
print(dictionary)  
# Output: {'name': 'Nikunj', 'age': 25, 'course': 'Data Analysis'}**

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

**my\_dict = {'a': 3, 'b': 1, 'c': 2}  
asc = dict(sorted(my\_dict.items(), key=lambda item: item[1]))  
desc = dict(sorted(my\_dict.items(), key=lambda item: item[1], reverse=True))  
print("Ascending:", asc)  
print("Descending:", desc)  
# Output: Ascending: {'b': 1, 'c': 2, 'a': 3}  
# Descending: {'a': 3, 'c': 2, 'b': 1}**

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

**dict1 = {1: 10, 2: 20}  
dict2 = {3: 30, 4: 40}  
dict3 = {5: 50, 6: 60}  
new\_dict = {\*\*dict1, \*\*dict2, \*\*dict3}  
print(new\_dict)  
# Output: {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}**

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

**my\_dict = {'a': 1, 'b': 2, 'c': 3}  
key = 'b'  
if key in my\_dict:  
 print(f"Key '{key}' exists in dictionary")  
else:  
 print(f"Key '{key}' does not exist")**

## **51) How Do You Traverse Through a Dictionary Object in Python?**

**my\_dict = {'a': 1, 'b': 2, 'c': 3}  
for key, value in my\_dict.items():  
 print(key, ":", value)**

## **52) How Do You Check the Presence of a Key in A Dictionary?**

**my\_dict = {'a': 1, 'b': 2, 'c': 3}  
print('a' in my\_dict) # True  
print('z' in my\_dict) # False**

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

**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.**

**my\_dict = {'a': 1, 'b': 2, 'c': 3, 'd': 4}  
keys = ['a', 'b']  
if all(key in my\_dict for key in keys):  
 print("All keys exist")  
else:  
 print("Some keys are missing")**

## **55) Write a Python script to merge two Python dictionaries.**

**dict1 = {'a': 1, 'b': 2}  
dict2 = {'c': 3, 'd': 4}  
merged = {\*\*dict1, \*\*dict2}  
print(merged)  
# Output: {'a': 1, 'b': 2, 'c': 3, 'd': 4}**

## **56) Write a Python program to map two lists into a dictionary.**

**from collections import Counter  
list1 = ['a', 'b', 'd', 'a', 'b', 'c']  
list2 = [400, 400, 400, 400, 400, 300]  
result = Counter(dict(zip(list1, list2)))  
print(result)  
# Output: Counter({'a': 400, 'b': 400, 'd': 400, 'c': 300})**

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

**my\_dict = {'a': 10, 'b': 50, 'c': 30, 'd': 70, 'e': 40}  
highest = sorted(my\_dict.values(), reverse=True)[:3]  
print("Highest 3 values:", highest)  
# Output: [70, 50, 40]**

## **58) Write a Python program to combine values in python list of dictionaries.**

**from collections import Counter  
data = [  
 {'item': 'item1', 'amount': 400},  
 {'item': 'item2', 'amount': 300},  
 {'item': 'item1', 'amount': 750}  
]  
result = Counter()  
for d in data:  
 result[d['item']] += d['amount']  
print(result)  
# Output: Counter({'item1': 1150, 'item2': 300})**

## **59) Write a Python program to create a dictionary from a string (count letters).**

**string = "w3resource"  
letter\_count = {}  
for ch in string:  
 letter\_count[ch] = letter\_count.get(ch, 0) + 1  
print(letter\_count)  
# Output: {'w': 1, '3': 1, 'r': 2, 'e': 2, 's': 1, 'o': 1, 'u': 1, 'c': 1}**

## **61) Write a Python function to calculate the factorial of a number.**

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

**return n \* factorial(n-1)  
 print(factorial(5)) # 120**

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

**def check\_range(num, start, end):  
 return start <= num <= end  
print(check\_range(5, 1, 10)) # True  
print(check\_range(15, 1, 10)) # False**

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

**def is\_perfect(n):  
 divisors = [i for i in range(1, n) if n % i == 0]  
 return sum(divisors) == n  
print(is\_perfect(28)) # True  
print(is\_perfect(12)) # False**

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

**def is\_palindrome(s):  
 return s == s[::-1]  
print(is\_palindrome("madam")) # True  
print(is\_palindrome("hello")) # False**

## **65) How Many Basic Types of Functions Are Available in Python?**

**1. Built-in Functions (e.g., len(), print(), max())  
2. User-defined Functions (created using def)  
3. Lambda (Anonymous) Functions (created using lambda)**

## **66) How can you pick a random item from a list or tuple?**

**import random  
items = [10, 20, 30, 40, 50]  
print(random.choice(items))**

## **67) How can you pick a random item from a range?**

**import random  
print(random.choice(range(1, 10)))**

## **68) How can you get a random number in python?**

**import random  
print(random.randint(1, 100)) # Random integer between 1 and 100  
print(random.random()) # Random float between 0 and 1**

## **69) How will you set the starting value in generating random numbers?**

**import random  
random.seed(10) # Set seed value  
print(random.random()) # Will give same result each time when seed is fixed**

## **70) How will you randomize the items of a list in place?**

**import random  
items = [1, 2, 3, 4, 5]  
random.shuffle(items)  
print(items)**

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

**The `open()` function is used to work with files in Python.  
Modes:  
- 'r' : Read  
- 'w' : Write (creates new file or overwrites)  
- 'a' : Append  
- 'b' : Binary  
- 'x' : Exclusive creation  
Example: f = open("test.txt", "w")**

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

**with open("test.txt", "r") as f:  
 data = f.read()  
 print(data)**

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

**with open("test.txt", "a") as f:  
 f.write("Hello World\n")  
  
with open("test.txt", "r") as f:  
 print(f.read())**

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

**n = 3  
with open("test.txt", "r") as f:  
 for i in range(n):  
 print(f.readline())**

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

**from collections import de ue  
n = 3  
with open("test.txt") as f:  
 for line in de ue(f, n):  
 print(line)**

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

**with open("test.txt") as f:  
 lines = f.readlines()  
print(lines)**

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

**with open("test.txt") as f:  
 data = f.read().splitlines()  
print(data)**

## **78) Write a Python program to find the longest words.**

**with open("test.txt") as f:  
 words = f.read().split()  
longest = max(words, key=len)  
print(longest)**

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

**with open("test.txt") as f:  
 count = sum(1 for line in f)  
print("Number of lines:", count)**

## **80) Write a Python program to count the fre uency of words in a file.**

**from collections import Counter  
with open("test.txt") as f:  
 words = f.read().split()  
print(Counter(words))**

## **81) Write a Python program to write a list to a file.**

**lines = ["Hello", "World", "Python"]  
with open("test.txt", "w") as f:  
 for item in lines:  
 f.write(item + "\n")**

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

**with open("test.txt", "r") as f1, open("copy.txt", "w") as f2:  
 for line in f1:  
 f2.write(line)**

## **83) Explain Exception handling? What is an Error in Python?**

**- Exception Handling allows you to handle runtime errors gracefully.  
- Errors can be Syntax Errors or Exceptions.  
- Syntax Errors: Occur due to wrong code structure.  
- Exceptions: Occur during execution (e.g., ZeroDivisionError, FileNotFoundError).**

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

**- A try block can have multiple except statements.  
Examples of exception classes:  
 ValueError, TypeError, KeyError, IndexError, ZeroDivisionError, FileNotFoundError**

## **85) When will the else part of try-except-else be executed?**

**- The else block executes only if no exception occurs in try block.**

## **86) Can one block of except statements handle multiple exception?**

**Yes, you can handle multiple exceptions in one block using a tuple:  
try:  
 ...  
except (ValueError, TypeError) as e:  
 print(e)**

## **87) When is the finally block executed?**

**- The finally block always executes regardless of exception occurrence.**

## **88) What happens when '1' == 1 is executed?**

**It returns False, because '1' is a string and 1 is an integer. Different data types.**

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

**try:  
 x = 10 / 0  
except ZeroDivisionError as e:  
 print("Error:", e)  
else:  
 print("No Error")  
finally:  
 print("Always executed")**

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

**try:  
 num = int(input("Enter an odd number: "))  
 if num % 2 == 0:  
 raise ValueError("This is not an odd number!")  
 print("You entered an odd number:", num)  
except ValueError as e:  
 print("Error:", e)**