Python Lab Preparation

✅ Keywords and Identifiers

📌 Viva Questions & Answers:

1. **What are keywords in Python? Give examples.**
    - Reserved words with special meaning. E.g., if, else, for, while, def, class, True, False, None.
2. **Can you use a keyword as a variable name?**
    - No, keywords are reserved.
3. **What is an identifier in Python?**
    - Names given to variables, functions, classes, etc.
4. **What are the rules for naming identifiers?**
    - Must begin with a letter (A–Z, a–z) or underscore (_). Can't use keywords. No special characters except _.

✅ Data Types

📌 Viva Questions & Answers:

1. **What are the basic data types in Python?**
    - int, float, str, bool, list, tuple, dict, set.
2. **Difference between list and tuple?**
    - List: mutable, uses [] — Tuple: immutable, uses ().
3. **Difference between set and dictionary?**
    - Set: unordered collection of unique items. Dictionary: key-value pairs.
4. **Which data types are mutable/immutable?**
    - Mutable: list, dict, set — Immutable: int, float, str, tuple, bool.
5. **How to check the type of a variable?**
    - Use type(variable_name).

👷 **Coding Questions:**

```
# 1. Type of user input
x = input("Enter something: ")
print("Type of input:", type(x))  # Always str

# 2. List and print elements
lst = [10, 20, 30, 40, 50]
print("Second element:", lst[1])
print("Last element:", lst[-1])

# 3. Tuple modification
t = (1, 2, 3)
# t[0] = 10  # Error: 'tuple' object does not support item assignment

# 4. Dictionary
student = {"Alice": 95, "Bob": 88}
print(student)

# 5. Set duplicates
s = {1, 2, 2, 3, 4}
print(s)  # {1, 2, 3, 4}
```

✅ **Type Conversion and Typecasting**

📌 **Viva Questions & Answers:**

1. **What is typecasting in Python?**
    - Converting one data type into another using built-in functions like int(), float(), str().

2. **Example of string to integer:**

```
s = "123"
n = int(s)
```

✅ **Input and Output Functions**

📌 **Viva Questions & Answers:**

1. **How do you take input in Python?**
    - Using input() function.
2. **What is the data type of input()?**
    - Always str.
3. **What does print("Hello", end=" ") do?**
    - Prints “Hello” and ends with a space instead of a newline.
4. What is f-string in python?
   Ans:
   An **f-string** in Python is a way to embed expressions inside string literals, using curly braces {}. It stands for **formatted string literal** and was introduced in **Python 3.6**.
5. **How can you format output using f-string?**

```
name = "Ariyan"
age = 20
print(f"My name is {name} and I am {age} years old.")
```

👷 **Coding Questions:**

```
# 1. Name and age
name = input("Enter your name: ")
age = input("Enter your age: ")
print(f"Hello {name}, you are {age} years old.")
```

```
# 2. Sum of two numbers
a = int(input("Enter first number: "))
b = int(input("Enter second number: "))
print(f"Sum: {a + b}")
```

✅ Comments in Python

📌 Viva Questions & Answers:

1. **How do you write single-line and multi-line comments?**
    - Single-line: # this is a comment
    - Multi-line:

```
"""
This is a
multi-line comment
"""
```

2. **Why are comments used in Python?**
    - To explain code for better understanding and maintenance.

✅ id() and type() Functions

📌 Viva Questions & Answers:

1. **What does id() function return?**
    - Memory address of the object.
2. **What does type() function return?**
    - The data type of the object.
3. **How to check if two variables refer to the same object?**
    - Use id() or the is operator.

Python Viva Questions and Coding Problems on Variables and Operators

---

Operators:-

Arithmetic Operators

- +, -, *, /, //, %, **

Viva Questions

1. **What is the difference between / and //?**
    - / gives float result, // gives integer (floor) result.
2. **What is the use of %?**
    - It gives the remainder of division.
3. **What is ** operator used for?**
    - Exponentiation (power).

Coding Questions

1. Write a program to find the quotient and remainder of two numbers.

```
a = 10
b = 3
print("Quotient:", a // b)
print("Remainder:", a % b)
```

Relational Operators

- ==, !=, >, <, >=, <=

Viva Questions

1. **What is the result of a relational operation?**
    - A boolean value: True or False.
2. **Can we compare strings using relational operators?**
    - Yes, based on lexicographical order.

Coding Questions

1. Write a program to check if a person is eligible to vote.

```
age = int(input("Enter age: "))
print("Eligible to vote?", age >= 18)
```

Logical Operators

- and, or, not

Viva Questions

1. **What are logical operators used for?**
    - To combine conditional statements.

Bitwise Operators

- &, |, ^, ~, <<, >>

Viva Questions

1. **What are bitwise operators used for?**
    - They operate on binary representations of integers.
2. **What is the result of 5 & 3?**
    - 1 (binary: 0101 & 0011 = 0001)

Assignment Operators:-

- =, +=, -=, *=, /=, //=, %=, **=

Viva Questions

1. **What is an assignment operator?**
    - It is used to assign values to variables.
2. **What is the difference between = and ==?**
    - = is assignment, == is comparison.

Coding Questions

1. Write a program to update the value of a variable using assignment operators.

```
x = 10
x += 5
print(x)
```

Identity Operators

- is, is not

Viva Questions

1. **What does is compare?**
    - It compares memory location (identity).

Coding Questions

```
a = [1, 2]
b = a
c = [1, 2]
print(a is b)  # True
print(a is c)  # False
```

Operator Precedence and Associativity

Viva Questions

1. **What is operator precedence?**
    - It determines the order in which operations are performed.

✅ **Python Conditional Statements - Viva & Coding Preparation**

🔹 **Theoretical Viva Questions with Answers**

---

2. What are the types of conditional statements in Python?

Answer:

- if statement
- if-else statement
- if-elif-else ladder
- Nested if statements

---

5. What is an elif in Python?

Answer: elif stands for 'else if'. It allows multiple conditions to be checked in sequence.

---

7. What is a nested if statement?

Answer: An if statement inside another if block is called a nested if statement.

- **Coding Questions (Other Than Assignment)**

Q2. Write a program to find the largest among three numbers.

```
a = int(input("Enter first number: "))
b = int(input("Enter second number: "))
c = int(input("Enter third number: "))
if a >= b and a >= c:
    print("Largest is", a)
elif b >= a and b >= c:
    print("Largest is", b)
else:
    print("Largest is", c)
```

Q3. Write a program to check whether a year is a leap year or not. (IMPORTANT)

```
year = int(input("Enter a year: "))
if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):
    print("Leap year")
else:
    print("Not a leap year")
```

Q4. Write a program to check whether a character is a vowel or consonant. (IMPORANT)

```
ch = input("Enter a character: ").lower()
if ch in 'aeiou':
    print("Vowel")
elif ch.isalpha():
    print("Consonant")
else:
    print("Not an alphabet")
```

Q5. Write a program to assign grades based on marks.

```
marks = int(input("Enter marks: "))
if marks >= 90:
    print("Grade A")
elif marks >= 75:
    print("Grade B")
elif marks >= 60:
    print("Grade C")
elif marks >= 40:
    print("Grade D")
else:
    print("Fail")
```

Python Viva and Coding Questions: Loops (for, while, break, continue, pass, nested loops)

---

1. Theoretical Viva Questions with Answers

Q1. What is a loop in Python?
A loop is a programming structure that repeats a sequence of instructions until a specific condition is met.

Q2. What are the types of loops in Python?

- for loop
- while loop

Q3. What is the difference between a for loop and a while loop?

- for loop is used when the number of iterations is known.
- while loop is used when the number of iterations is not known and depends on a condition.

Q4. What is the use of the break statement?
The break statement is used to exit a loop prematurely when a certain condition is met.

Q5. What is the use of the continue statement?
The continue statement skips the current iteration and continues with the next iteration of the loop.

Q9. What happens if the condition in a while loop is always true?
It creates an infinite loop unless a break statement is used to exit it.

---

2. Coding Questions with Answers

Q1. Calculate the sum of numbers from 1 to 100 using a while loop.

```
i = 1
sum = 0
while i <= 100:
    sum += i
    i += 1
print("Sum:", sum)
```

Q2. Print multiplication table of a number using for loop.

```
num = int(input("Enter a number: "))
for i in range(1, 11):
    print(f"{num} x {i} = {num*i}")
```

Python Viva and Coding Questions: Functions

---

1. Defining and Calling Functions

Q1. What is a function in Python?
A function is a block of reusable code that performs a specific task.

Q2. How do you define a function in Python?
Using the def keyword:

```
def greet():
    print("Hello")
```

Q3. How do you call a function?

```
greet()  # Output: Hello
```

---

2. Parameters and Return Values

Q4. What is the difference between parameters and arguments?
Parameters are variables in the function definition, arguments are values passed during the function call.

Q5. Can a function return a value? Give an example.

```
def add(a, b):
    return a + b


print(add(5, 3))  # Output: 8
```

---

4. Recursion

Q9. What is recursion?
When a function calls itself to solve a smaller part of the problem.

Q10. Write a recursive function to calculate factorial.

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

```
    return n * factorial(n - 1)


print(factorial(5))  # Output: 120
```

---

5. Built-in Functions

Q11. What does len() do?
Returns the length of a sequence.

```
len("hello")  # Output: 5
```

Q12. What does range() do?
Generates a sequence of numbers.

```
list(range(1, 6))  # Output: [1, 2, 3, 4, 5]
```

Q13. What does sum() do?
Returns the sum of all elements in an iterable.

```
sum([1, 2, 3])  # Output: 6
```

Function	**Job (Description)**
abs(x)	Returns the absolute value of a number
all(iterable)	Returns True if all elements in the iterable are true
any(iterable)	Returns True if any element in the iterable is true
bin(x)	Converts an integer to a binary string
bool(x)	Converts a value to Boolean (True or False)
chr(x)	Converts an ASCII code (integer) to a character
ord(x)	Converts a character to its ASCII code
complex(a, b)	Returns a complex number (a + bj)
dict()	Creates a dictionary
divmod(a, b)	Returns a tuple (quotient, remainder)
enumerate()	Adds index to iterable (returns enumerate object)
eval(expr)	Evaluates a Python expression from a string

Function	Job (Description)
float(x)	Converts x to a float
format()	Formats strings
input()	Takes user input as string
int(x)	Converts x to an integer
len(x)	Returns the length (number of items)
list()	Creates a list
max(iterable)	Returns the largest item
min(iterable)	Returns the smallest item
pow(x, y)	Returns x to the power of y
print()	Prints output to the screen
range(start, stop, step)	Generates a sequence of numbers
reversed()	Returns reversed iterator
round(x)	Rounds a number to nearest integer or specified decimals
set()	Creates a set (collection of unique elements)
sorted()	Returns a sorted list
str()	Converts a value to string
sum()	Returns the sum of all items in an iterable
tuple()	Creates a tuple
type()	Returns the type of an object
zip()	Combines multiple iterables element-wise

Coding Questions (Functions Related)

Q14. Write a function to check whether a number is even or odd.

```
def is_even(n):
    return n % 2 == 0
```

```
print(is_even(4))  # Output: True
```

Q16. Write a function that returns the largest among three numbers.

```
def largest(a, b, c):
    return max(a, b, c)

print(largest(10, 20, 15))  # Output: 20
```

Python Viva & Coding Questions: Strings Topic

Viva Questions (With Answers)

1. What is a string in Python?

A string in Python is a sequence of characters enclosed within single quotes (' '), double quotes (" "), or triple quotes (''' ''' or """ """).

2. How are strings indexed in Python?

Strings are indexed using zero-based indexing. That means the first character has index 0, the second has index 1, and so on. Negative indexing is also allowed (e.g., -1 refers to the last character).

3. What is string slicing?

Slicing is used to access a portion (substring) of a string using the syntax: string[start:stop:step]. **(IMPORTANT)**

4. Are strings mutable in Python?

No, strings in Python are immutable. Once created, their content cannot be changed.

5. What is the difference between is and == in string comparison?

== compares the values of two strings, while is checks if two strings refer to the same memory location (identity).

6. Name some commonly used string methods.(IMP)

lower(), upper(), replace(), find(), count(), strip(), split(), join(), startswith(), endswith().

7. How can you reverse a string in Python?

Using slicing: s[::-1]

8. What is a palindrome?

A palindrome is a string that reads the same forwards and backwards, e.g., "madam" or "racecar".

9. How do you iterate through each character in a string?

Using a for loop: for char in string:

10. What is the output of 'hello'.upper()?

HELLO

12. How to find if a substring exists within a string?

Using in operator or find() method.

Coding Questions (With Answers)

3. Write a program to count vowels and consonants in a string.

```
s = input("Enter a string: ").lower()
vowels = "aeiou"
vowel_count = 0
consonant_count = 0
for char in s:
    if char.isalpha():
        if char in vowels:
            vowel_count += 1
        else:
            consonant_count += 1
print("Vowels:", vowel_count)
print("Consonants:", consonant_count)
```

7. Write a program to find the frequency of each character in a string.

```
s = input("Enter a string: ")
freq = {}
```

```
for char in s:
    freq[char] = freq.get(char, 0) + 1
print("Character frequencies:")
for char, count in freq.items():
    print(char, ":", count)
```

8. Write a program to check if a string contains only digits.

```
s = input("Enter a string: ")
if s.isdigit():
    print("The string contains only digits")
else:
    print("The string contains characters other than digits")
```

Python Viva and Coding Questions: Lists and Arrays

Viva Questions with Answers

1. What is a list in Python?

A list is a built-in data type in Python used to store multiple items in a single variable. It is mutable and can contain elements of different data types.

2. How do you create a list in Python?

```
my_list = [1, 2, 3, 4, 5]
```

3. How do you access elements in a list?

By using indexing. Index starts from 0.

```
print(my_list[0])  # Output: 1
```

4. What are some common list methods?

- append(item): Adds item at the end
- remove(item): Removes first occurrence of item
- pop(index): Removes item at index (last item if index not given)
- sort(): Sorts list in ascending order
- reverse(): Reverses the order of list

5. What is the difference between pop() and remove()?

- pop() removes an item at a specific index (or the last item if no index is specified).
- remove() removes the first matching value.

6. How do you traverse a list?

```
for item in my_list:
    print(item)
```

7. What is the difference between list and array?

In Python, lists can store elements of different data types, while arrays (using modules like array or NumPy) store elements of the same data type and are more memory-efficient.

8. Can lists be nested in Python?

Yes, lists can contain other lists as elements (nested lists).

```
nested_list = [[1, 2], [3, 4]]
```

9. Is a list mutable?

Yes, elements of a list can be changed after creation.

10. How do you find the length of a list?

Using len() function:

```
len(my_list)
```

Python Viva and Coding Questions: Pattern Printing Using Nested Loops

---

II. Coding Questions : (IMPORTANT)

1. **Print a right-angled triangle of stars:**

```
n = 5
for i in range(1, n+1):
    for j in range(i):
        print('*', end=' ')
```

```
    print()
```

2. **Print a square of stars:**

```
n = 4
for i in range(n):
    for j in range(n):
        print('*', end=' ')
    print()
```

3. **Print a right-aligned triangle of stars:**

```
n = 5
for i in range(1, n+1):
    print(' ' * (n-i) + '*' * i)
```

4. **Print a pyramid pattern:**

```
n = 5
for i in range(n):
    print(' ' * (n-i-1) + '* ' * (i+1))
```

5. **Print a number triangle:**

```
n = 5
for i in range(1, n+1):
    for j in range(1, i+1):
        print(j, end=' ')
    print()
```

6. **Print a reverse number triangle:**

```
n = 5
for i in range(n, 0, -1):
    for j in range(1, i+1):
        print(j, end=' ')
    print()
```

7. **Print an inverted star pyramid:**

```
n = 5
```

```
for i in range(n, 0, -1):
    print(' ' * (n-i) + '* ' * i)
```

8. **Print pattern with ASCII values (A to Z triangle):**

```
n = 5
for i in range(65, 65+n):
    for j in range(65, i+1):
        print(chr(j), end=' ')
    print()
```

Python Viva and Coding Preparation Guide

✅ **Theoretical Viva Questions with Answers**

1. What is the difference between list and tuple?

- **List:** Mutable, enclosed in square brackets [], can be changed.
- **Tuple:** Immutable, enclosed in parentheses (), cannot be changed.

2. What is the difference between is and == in Python?

- == checks for value equality.
- is checks for reference (memory location) equality.

5. What are mutable and immutable types?

- **Mutable:** Can be changed after creation (e.g., list, dict, set)
- **Immutable:** Cannot be changed (e.g., int, float, string, tuple)

7. Explain local vs global variables.

- **Local variable:** Defined inside a function and accessible only within that function.
- **Global variable:** Defined outside functions and accessible globally.

LAB QUESTIONS ----------IMP----------

1. write a program in python to swap two numbers using *3rd* variable

```
a=10
b=5
c=a
a=b
b=c
print(a)
print(b)
```

2. write a program in python to swap two variables without using *3rd* variable

```
a=int(input("Enter the number 1: "))
b=int(input("Enter the number 2: "))

a=a+b
b=a-b
a=a-b
print(f"The value of a is {a} and the value of b is {b} after swapping")
```

3. write a program to find smallest among n numbers

```
a=[2,3,6,8]
smallest=a[0]
for i in range(1,len(a)):
   if a[i]<smallest:
      smallest=a[i]
print(f"The smallest number is {smallest}")
```

// means floor division but / means float division that will return folating number means exact ans , and floor division will return nearest whole num

#how to take user input in python as a for loop (special)

```
n=int(input("Enter how many numbers you want in the array: "))
a=[]
for i in range(n):
  a.append(int(input(f"Enter the element {i+1}: ")))
print(a)
```

4. Write a program to find area of triangle :

```
base=int(input("Enter the base of triangle: "))
height=int(input("Enter the height of triangle: "))
area=(base*height)/2
print(f"The area of triangle is {area}")
```

5. write a program to convert distance a.meter to km b.km to meter

```
a=int(input("Enter the distance in meter: "))
print(f"The distance in km is {a/1000}")
b=int(input("Enter the distance in km: "))
print(f"The distance in meter is {b*1000}")
```

6. write a program to find simple interest and take all variables as user input

```
p=float(input("Enter the principal(Rs): "))
r=float(input("Enter the rate(Percentage): "))
t=int(input("Enter the time(Year): "))
si=(p*r*t)/100
print(f"The simple interest for time {t} is {si}")
```

7. write a program to find area of circle

```
r=int(input("Enter the radius of circle: "))
area=3.14*(r**2)
print(f"The area of circle is {area}")
```

8. For a given string ‘bamboozled’ write a program to print the following output

i. ba ii. ed iii. mboozled (3 times) iv. bambo (4 times)

v. delzoobmab vi. bamboozled vii. bmoze viii. bbzd (2 times) ix. boe

```
a="bamboozled"
print(a[0:2])
print(a[-2:])
for i in range(3):
    print(a[2:])
for i in range(4):
    print(a[:5])
print(a[::-1])
print(a[::])
print(a[::2])
for i in range (2):
    print(a[::3])

print(a[::4])
```

9) write a program to check whether a user input number is even or odd

```
num=int(input("Enter the number: "))
```

```
if num%2==0:
   print("The number is even")
else:
   print("The number is odd")
```

10) write a program to check a user input number is divisible by 7 or not

```
num=int(input("Enter the number: "))
if num%7==0:
   print("The number is divisible by 7")
else:
   print("The number is not divisible by 7")
```

11) write a program to check a string is palindrome or not

```
a=input("Enter the string: ")
b=a[::-1]
if a==b:
   print("The string is palindrome")
else:
   print("The string is not palindrome")
```

12) write a program to remove a letter from a string given by the user:

```
a=input("Enter the string: ")
b=input("Enter the letter you want to remove: ")
c=input("Enter the letter you want to replace: ")
print(a.replace(b,c))
```

if u remove the letter only , rplace by "";

13) write a program to print even length words in a string:

```
a=input("Enter the string: ")
b=a.split()
print("The even length words are: ")
for i in b:
    if len(i)%2==0:
        print(i)
```

14) Write a program to print the following pattern

```
*
* *
* * *
* * * *
* * *
* *
*
```

```
n=int(input("Enter the number of lines: "))
for i in range(1,n+1):
    print("* "*i)
for i in range(n-1,0,-1):
    print("* "*i)
```

15) write a program that will print the number of digits and number of alphabets in a string

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

```
b=0
c=0
for char in a:
    if char.isalpha():
        c+=1
    elif char.isdigit():
        b+=1
print(f"The number of digits is {b}")
print(f"The number of alphabets is {c}")
```

16)write a program to print all prime numbers form 1 to 100

```
import math
def isPrime(num):
    for i in range(2,int(math.sqrt(num))+1):
        if num%i==0:
            return False
    return True

for i in range(2,101):
    if isPrime(i):
        print(i)
```

17)write a program to find factorial of any number entered by user input

```
def factorial(num):
    if num==0:
        return 1
    else:
        return num*factorial(num-1)
```

```
num=int(input("Enter the number: "))
print(factorial(num))
```

18)write a program to find if the area of a rectangle is greater than its perimeter

```
l=int(input("Enter the length: "))
b=int(input("Enter the breadth: "))
if l*b>2*(l+b):
    print("The area of rectangle is greater than its perimeter")
else:
    print("The area of rectangle is not greater than its perimeter")
```

19)write a program to check whether a triangle is valid or not

```
a=int(input("Enter the length of side a: "))
b=int(input("Enter the length of side b: "))
c=int(input("Enter the length of side c: "))
if a+b>c and b+c>a and c+a>b:
    print("The triangle is valid")
else:
    print("The triangle is not valid")
```

20)write a program to find LCM of two numbers

```
a=int(input("Enter the number 1: "))
b=int(input("Enter the number 2: "))
if a>b:
    c=a
else:
```

```
    c=b
while True:
    if c%a==0 and c%b==0:
        print(f"The LCM of {a} and {b} is {c}")
        break
    c+=1
```

21)write a program to find GCD of two numbers

```
a=int(input("Enter the number 1: "))
b=int(input("Enter the number 2: "))
if a<b:
    c=a
else:
    c=b
while True:
    if a%c==0 and b%c==0:
        print(f"The GCD of {a} and {b} is {c}")
        break
    c-=1
```

22)write a program to check if a number is perfect or not :

```
n=int(input("Enter the number: "))
sum=0
for i in range(1,n):
    if n%i==0:
        sum+=i
if sum==n:
```

```
    print("The number is perfect")
else:
    print("The number is not perfect")
```

23)write a program to find sum of all elements in an array:

```
n=int(input("Enter the number of elements: "))
a=[]
for i in range(n):
    a.append(int(input(f"Enter the element {i+1}: ")))
sum=0
for i in a:
    sum+=i
print(f"The sum of all elements is {sum}")
```

24)write a program to reverse elements of an array

```
n=int(input("Enter the number of elements: "))
a=[]
for i in range(n):
    a.append(int(input(f"Enter the element {i+1}: ")))
a.reverse()
print(a)
```

25)write a program to find a specific element in an array and return it's index

1.Linear Search:

```
n=int(input("Enter the number of elements: "))
a=[]
```

```
for i in range(n):
    a.append(int(input(f"Enter the element {i+1}: ")))
x=int(input("Enter the element you want to find: "))
for i in range(n):
    if a[i]==x:
        print(f"The index of {x} is {i}")
```

2.Binary Search:

```
n=int(input("Enter the number of elements: "))
a=[]
for i in range(n):
    a.append(int(input(f"Enter the element {i+1}: ")))
a.sort()
x=int(input("Enter the element you want to find: "))
low=0
high=n-1
while low<=high:
    mid=(low+high)//2
    if a[mid]==x:
        print(f"The index of {x} is {mid}")
        break
    elif x<a[mid]:
        high=mid-1
    else:
        low=mid+1
```

25)Bubble Sort:

```
a=[35,76,21,3,89,65]
```

```
for i in range(len(a)-1):
    for j in range(len(a)-i-1):
        if a[j]>a[j+1]:
            temp=a[j]
            a[j]=a[j+1]
            a[j+1]=temp
print(a)
```

26)Insertion Sort:

```
a=[35,76,21,3,89,65]
for i in range(1,len(a)):
    key=a[i]
    j=i-1
    while j>=0 and key<a[j]:
        a[j+1]=a[j]
        j-=1
    a[j+1]=key
print(a)
```