1.What are Logical Operators? how many are they ?

Ans: Logical operators are words like and, or, and not that helps us make decisions in programs.

They are used to combine conditions(True or False) and get a final answer (also True or False).

There are 3 logical operators in python:

and – True only if both conditions are True

or – True if at least one condition is True

not – Opposite of the condition(True becomes False, False becomes True)

2.what is d/f b/w the logical AND & logical OR ?

Ans: Logical AND (and)

- Returns True only if both conditions are True
- If any one is False, the result is False

```
age = 20
has_id = True

if age > 18 and has_id:
    print("You can enter.")
else:
    print("Entry denied.")
```

Logical OR (or)

- Returns True if at least one condition is True
- Returns False only if both conditions are False

```
age = 16
has_pass = True

print(age > 18 or has_pass)
```

3.what are membership operators ? how many are they ?

Ans: The membership operators in Python help us determine whether an item is present in a given group like a list, string, or set.

They answer:

➡ **"Is this item present?"**

➡ **"Is this item not present?"**

There are **2 membership operators** in Python:

in ----> True if value is present

Not in --> True if value is Not Present

4.what is d/f b/w in and not in operators?

Ans: **in operator:**

- Checks if a value is present in a list, string, or any collection.
- Return True if it exists, otherwise False.
  Example:
  fruits = ["apple", "banana", "mango"]
  print("apple" in fruits) #True

not in operator:

- Checks if a value is Not present in a list, string, or collection.
- Return True if it doesn't exist, otherwise False.
  Example:
  colors = ["red", "blue", "green"]
  print("yellow" not in colors) #True

5. what is the d/f b/w == and != operators ?

Ans: ==(Equal to)

- Checks if two values are the same
- Returns True if they are equal, else False
  Example:
  a = 5
  print(a == 5) #True

!=(Not equal to)

- Checks if two values are different
- Return True if they are not equal, else False
  Example:
  a = 5
  print(a != 3) # True

6. what are conditional statements in python ? write a syntax and simple example ?

Ans: Conditional statements are used to make decisions in a program. They check if a condition is True or False, and based on that, they run certain code.

Types of conditional statements:

1.if statement

Syntax: if condition:

```
    #code run if condition is true
```

Example:

```
age = 18
if age >= 18:
   print("You can vote.")
```

2. if-else statement

Syntax: if condition:

```
        # code if True
    else:
        # code if False
```

Example:

```
marks = 40
if marks >= 35:
   print("Pass")
else:
   print("Fail")
```

3. if-elif-else statement

```
if condition1:
   # code if condition1 is True
elif condition2:
   # code if condition2 is True
else:
   # code execute if all are False
```

Example:

```
num = 0
if num > 0:
```

```
    print("Positive")
elif num < 0:
    print("Negative")
else:
    print("Zero")
```

7. write a program to demonstrate the if-else conditions ?

```
  num = int(input(“Enter a number: “))  # 52
   if num % 2 == 0:     # 52 %2 ==0
       print(“The number is Even”)
else:
       print(“The number is odd”)
```

8. write if-else-if-else ladder with a simple example ?

```
 marks = int(input(“Enter your marks: “))
if marks>=92:
     print(“A Grade”)
elif marks >= 75:
      print(“B Grade”)
elif marks >= 60:
       print(“C Grade”)
elif marks >= 35:
       print(“D Grade”)
else:
       print(“Fail”)
```

9.write a program to demonstrate how nested conidtions works in python ?

```
if marks >= 35:
       if marks >= 92 and marks <= 100:
            print(“A+ Grade”)
      elif marks >=71 and marks <92:
            print(“B Grade”)
```

```
   elif marks >= 51 and marks < 71:
        print(“C Grade”)
  elif marks >= 35 and marks < 51:
       print(“Just pass”)
 else:
       print(“Please enter marks upto 100 only”)
else:
 print(“You Failed”)
```

10.what is indentation in python ? what is importance of it in python ? explain with an example?

Ans: Indentation means adding spaces or tabs at the beginning of a line of code to show which block it belongs to.

In Python, indentation is not optional — it's a must.
It tells Python which lines of code belong together.

- Python uses indentation to group lines of code (like inside if, for, while, functions, etc.)
- Without proper indentation, your program will give an error
- It improves readability and structure

```
age = 20
if age >= 18:
   print("You are an adult.")  # This line is inside the 'if' block
   print("You can vote.")
```

output:

```
You are an adult.
You can vote.
```

```
age = 20
if age >= 18:
print("You are an adult.") # error: no indentation
```

11.what is error and how many type of errors do you know?

An **error** is something that **goes wrong in a program** and **stops it from running properly**.

Errors happen when:

- You write wrong code (like a typo)
- You try to do something not allowed (like dividing by zero)
- You use something that doesn't exist

1. Syntax Error

- You wrote code that breaks Python's rules.
- Happens when code is not written correctly.

print("Hello" # Missing closing bracket

Error: syntax error

2. Runtime Error

- The code is correct, but an error happens while the program is running.

Example:
a = 5/0 # you can't divide by zero

Error: ZeroDivisionError

3. Logical Error

- Code runs without errors, but the output is wrong because the logic is incorrect.

num = 4

print("Square:", num + num)

No error shown, but result is **wrong**

12.write an example each to demonstrate syntax error and name Error and keyError ?

1. SyntaxError

Occurs when you break Python's grammar rules, like missing punctuation or indentation.

Example:

print("Hello, world" # Syntax Error

SyntaxError: unexpected EOF while parsing

2. NameError

Happens when you use a variable or function name that doesn't exist or is not defined.

print(age) # 'age' is not defined

NameError: name 'age' is not defined

3. KeyError

Occurs when you try to access a key in a dictionary that doesn't exist.

```
student = {"name": "Alice", "age": 20}

print(student["grade"])   #  'grade' key not in dictionary
```

KeyError: 'grade'

13.what is loop and how many types of loops are there in python?

A loop is used to repeat a block of code again and again — as long as a condition is true or for a fixed number of times.

Example:
If you want to print your name 5 times, instead of writing 5 print() statements, you use a loop.

For loop ---> Repeats for a fixed number of times.

While loop ----> Repeats while a condition is true.

1. **for Loop** – Use when you know how many times to repeat
   ```
   for i in range(5):
       print("Hello")
   ```
2. **while Loop** – Use when you don't know exactly how many times, but want to repeat until a condition becomes False

14. write an example for for loop using list?

```
fruits = ["apple", "banana", "mango", "orange"]

for fruit in fruits:

    print(fruit)
```

output:

apple

banana

mango

orange

15. write an example for for loop using str and dict and tuple ?

For loop with a String

```
name = "nikki"
```

```
for char in name:
    print(char, end=” “)
# n i k k i
```

For loop with a Tuple

```
numbers = (10, 20, 30, 40)
for num in numbers:
     print(num)
```

For loop with a Dictionary

```
student = {"name": "nikki", "age": 20, "grade": "A"}
for i in student:
    print(i, ":", student[i])
```