

1. Explain programming and python in detail to write.
- Programming -  
programming is the process of designing, writing, testing, and maintaining a set of instructions called a program that tells a computer how to perform a specific task. These instructions are written using a specific programming language. The main purpose of programming is to solve problems, automate tasks, and develop applications such as websites, software and mobile apps.

• Python -  
Python is a high-level, interpreted, and general-purpose programming language. It was created by Guido Van Rossum and released in 1991. Python is known for its simple syntax and easy readability.

- Characteristics of Python :
  - 1. Python is very easy to learn and use
  - 2. Simple and readable Syntax.
  - 3. It is a Interpreted language.
  - 4. Python is object-oriented and procedural
  - 5. It is also platform Independent.
  - 6. It has Large Standard Library.
- Applications of Python :  
The Applications of Python are -

- 1. Web development
- 2. Data Science and Machine Learning
- 3. Artificial Intelligence
- 4. Automation and Scripting
- 5. Software development
- 6. Game Development

## • Types of Comments in python:

### 1. Single Line Comment:

It has only one line.

Ex: `# This is a single-line comment`

### 2. Multi-line Comment:

It has so many lines.

Ex: `## This is a multi-line comment`

This is a multi-line comment. It consists of three lines.

## • Importance of python in Modern Software Development

python programming language is widely used because it increases productivity, reduces development time, and supports modern technologies like AI, ML, and data analytics. Many companies use python for real-world applications.

## • Describe Data types and operators in python with suitable examples.

### → Built-in Data types in python

Python provides several built-in data types which are classified as follows:

#### 1. Numeric Data types:

Numeric Data types are used to store numerical values.

→ int - stores whole numbers

→ float - stores decimal number

→ Complex - stores complex numbers

examples: `a = 10`

`b = 3.5`

`c = 2 + 3j`

## 2. Sequence Data Types :

Sequence data types store multiple values in an ordered manner.

- string (str) - stores text form
- List - stores mutable ordered elements
- tuple - stores immutable ordered elements

example : name = "python"

numbers = [1, 2, 3, 4]

t = (10, 20, 30, 40)

## 3. Set Data type :

A set stores unique and unordered elements.

example: s = {1, 2, 3, 4}

## 4. Mapping Data type :

mapping data type stores data in key-value pairs.

### • Dictionary (dict)

example:

Student = {"name": "manu", "age": 20}

## 5. Boolean Data Type :

Boolean data type stores only two values : True or False.

Example:

result = True

→ Type Identification using type() function:

The type() function is used to identify the data type of a variable.

example: x = 10

print(type(x))

These are the different Built-in data types in Python.

## • Python Operators :-

Operators are special symbols used to perform operations on variable and values.

### 1. Arithmetic Operators:

Used to perform mathematical operations.

operator operation

+ Addition

- Subtraction

\* Multiplication

/ Division

% Modulus

example:  $a = 10$

$b = 5$

`print(a + b)`

`print(a * b)`

### 2. Assignment Operators:

Used to assign values to variables.

example: operator Description Example

`a =` Assigns value  $a = 10$

`+ =` Add and assign  $a + = 5$

`- =` Subtract and assign  $a - = 2$

### 3. Comparison Operators:

comparison operators are used to comparison operators is either True or False, are used to compare two values.

operator Description

`==` Equal to

Example

$a == b$

`!=` Not equal to

$a != b$

`>` Greater than

$a > b$

operator	Description	Example
$<$	Less than	$a < b$
$\geq$	Greater than or equal to	$a \geq b$
$\leq$	Less than or equal to	$a \leq b$

ex:  $a = 10$   
 $b = 5$   
print ( $a \geq b$ )

#### 4. Logical operators:

It is used to combine conditional statements.

operators : and, or, not

example:  $a = \text{True}$   
 $b = \text{False}$   
print ( $a \text{ and } b$ )

#### 5. Membership operators:

Used to check whether a value exists in a sequence

operators : in, not in

example:  $\text{nums} = [1, 2, 3]$   
print ( $2 \text{ in } \text{nums}$ )

#### 6. Identity operators:

used to compare memory location of objects.

operators : is, is not

example:  $a = 10$   
 $b = 10$   
print ( $a \text{ is } b$ )

#### • Real-world usage of Operators:

Operators are used in calculations, decision-making, comparisons, and data processing in real-life applications like billing systems and result evaluation.

Explain python input and output operations in detail.

Input and output operations in python are used to take data from the user and display results on the screen. Input allows the user to enter values, while output displays processed information.

### • Input() Function and its Default Data type

The input() function is used to take input from the user during program execution. By default, the input taken using input() is of String data type, even if the user enters numbers.

example: name = input("Enter your name: ")

### • Type Conversion while taking Input:

Since input is always a String, type conversion is required to perform mathematical operations. Functions like int(), float(), and str() are used for type conversion.

example: age = int(input("Enter age: "))  
salary = float(input("Enter salary: "))

### • Taking Multiple inputs :

python allows taking multiple inputs from the user in a single line using the split() function.

example: a, b = input("Enter two numbers: ").split()  
To convert them into integers:  
a, b = map(int, input("Enter two numbers: ").split())

- Formatted Output Using `print()`, Separators and Format Specifiers

### - `print()`

The `print()` function is used to display output on the screen.

### - Using Separators

The `sep` parameter is used to separate values while printing.

Example: `print(a, b, sep = " , ")`

### - Using Format Specifiers

Format Specifiers are used to control the output format.

- `%d` → integer
- `%f` → float
- `%s` → string

ex: `print("Age is %d" % age)`

## \* Discuss Control Statements and Decision-Making Statements in Python.

Control statements in python are used to control the flow of execution of a program. They determine which statements should be executed and in what order. Control statements help in making decisions, repeating tasks, and managing the execution flow of a program effectively.

### • Importance of Control statements

Control statements make programs logical, flexible, and meaningful. They allow the program to execute different blocks of code based on conditions and requirements instead of executing all statements sequentially.

- Types of Control Statements in Python.  
Based on their functionality, control statements are classified into three types:
  1. Decision-Making statements:  
Decision-making statements are used to make decisions based on conditions. The condition is evaluated as either True or False, and accordingly a particular block of code is executed. Examples include if, if-else, and if-elif-else.
  2. Looping statements:  
Looping statements are used to execute a block of code repeatedly until a given condition is satisfied. They help in reducing code repetition and improving program efficiency. Examples include for loop and while loop.
  3. Jump statements:  
Jump statements are used to transfer control from one part of the program to another. They are used to stop, skip, or pass the execution of statements. Examples include break, continue and pass.
- Decision-Making statements  
Decision-making statements play an important role in controlling the execution of a python program. They execute different blocks of code based on conditions.
- 1. if statement:  
The if statement executes a block of code only when the given condition is true.

Ex:  $a = 10$

$b = 5$

if  $a > b$ :

    print ("A is greater")

## 2. if-else statement:

The if-else statement executes one block of code if the condition is true and another block if the condition is false.

Ex:  $num = 7$

if  $num \% 2 == 0$ :

    print ("Even number")

else:

    print ("Odd number")

## 3. if-elif-else statements

The if-elif-else statement is used to check multiple conditions.

Ex:  $marks = 80$

if  $marks >= 90$ :

    print ("Grade A")

elif  $marks >= 75$ :

    print ("Grade B")

else:

    print ("Grade C")

## • Syntax Flow and Execution Control

In python decision-making statements, the execution of the program depends on the condition provided.

• First the Condition is evaluated.

• If the Condition is True, the corresponding block is executed.

- If the condition is false, control moves to the ~~if~~ ~~elif~~ or ~~else~~ block.
- If the condition is false, control directly moves to the ~~else~~ block.
- Only one block of code is executed at a time.
- Python uses indentation to control the execution flow.

Ex:

```
x = 5
y = 10
if x > y :
    print ("x is greater")
else:
    print ("y is greater")
```

Write an Essay on python programming Fundamentals.

python programming fundamentals form the foundation of learning and developing software applications. programming is the process of writing instructions that a computer follows to perform specific tasks.

python is a high-level, interpreted programming language known for its simple and readable syntax. It uses indentation instead of brackets, which improves readability and reduces syntax errors. this makes python suitable for beginners as well as professionals.

variables in python are used to store data, and python provides various built-in data types such as integers, floats, strings, lists, tuples, sets, and dictionaries. operators are used to perform operations like arithmetic, comparisons & logical decisions.

Input and output operations allow interaction between the user and the program. The `input()` function is used to take input from the user, and the `print()` function is used to display output.

Control flow statements such as `if`, `if-else`, and `if-elif-else` help in decision making. These statements allow the program to execute different blocks of code based on conditions.

Python fundamentals are widely used in real-world applications such as web development, data science, artificial intelligence, automation, and software development. Due to its simplicity and powerful features, Python has become one of the most popular programming languages.

1. Movie Ticket pricing

```
age = int(input("Enter age: "))
movie_type = input("Is it 3D? (yes/no): ")
if age < 13:
    price = 150
else:
    price = 250
if movie_type == "yes":
    price += 50
print("Ticket price: ", price)
```

2. College Attendance Rule

```
attendance = int(input("Enter attendance percentage: "))
medical_cert = input("Do you have a medical certificate? (yes/no): ")
if attendance >= 75:
    print("Allowed to write exam")
elif attendance >= 60 and medical_cert == "yes":
    print("Allowed to write exam")
else:
    print("Not allowed to write exam")
```

3. E-commerce Discount

```
bill_amt = int(input("Enter bill amount: "))
is_prime = input("Are you a prime member? (yes/no): ")
if bill_amt >= 5000:
    discount = 20
elif bill_amt >= 2000:
    discount = 10
else:
    discount = 0
if is_prime == "yes":
    discount += 5
```

```
final_amt = bill_amt - (bill_amt * discount / 100)
print ("Final amount: ", final_amt)
```

#### 4. Smartphone Battery Warning

```
battery = int(input("Enter battery percentage: "))
is_charging = input("Is your phone charging? (Yes/no): ")
if is_charging == "Yes":
    print("charging")
elif battery <= 20:
    print("Low battery")
elif battery > 80:
    print("Full")
else:
    print("Normal")
```

#### 5. Driving License Check

```
age = int(input("Enter age: "))
test_passed = input("Have you passed the driving test? (Yes/no): ")
if age >= 18 and test_passed == "Yes":
    print("Eligible for license")
else:
    print("Not eligible for license")
```

#### 6. Online Food Delivery

```
order_amt = int(input("Enter order amount: "))
is_gold = int(input("Are you a gold member? (1/0): "))
distance = int(input("Enter distance(km): "))
if distance > 10:
    print("Delivery charged")
elif order_amt >= 500 or is_gold == 1:
    print("Free delivery")
```

else:  
print("Delivery charged")

7. Bank Loan Approval  
Salary = int(input("Enter salary:"))  
Credit\_Score = int(input("Enter credit score:"))  
if (salary >= 30000 and credit\_score >= 700):  
 print("Loan Approved")  
else:  
 print("Loan Rejected")

8. Electricity Bill  
units = int(input("Enter units consumed:"))  
bill = 0  
if units > 200:  
 bill += 100 \* 2 + 100 \* 3 + (units - 200)  
elif units > 100:  
 bill += 100 \* 2 + (units - 100) \* 3  
else:  
 bill += units \* 2  
print("Bill amount: ", bill)

9. Student Scholarship  
marks = int(input("Enter marks:"))  
income = int(input("Enter family income:"))  
single\_parent = int(input("Single parent? (1/0):"))  
if marks >= 85 and (single\_parent == 1 or income < 500000):  
 print("scholarship granted")  
else:  
 print("scholarship denied")

## 10 Online Exam Result

```
theory = int(input("Enter theory marks: "))
practical = int(input("Enter practical marks: "))
total = theory + practical
if (theory >= 40 and practical >= 40) or (total >= 100):
    print("pass")
else:
    print("fail")
```

## 11 Hotel Room Pricing

```
is_weekend = int(input("Is weekend? (1/0): "))
days_stayed = int(input("Enter days stayed: "))
rate = 4000 if is_weekend == 1 else 3000
bill = days_stayed * rate
if days_stayed > 3:
    bill *= 0.85
print("Final bill:", bill)
```

## 12 Gaming Level Unlock

```
score = int(input("Enter Score: "))
is_premium = int(input("premium pass? (1/0): "))
used_cheat = int(input("Used cheat? (1/0): "))
if used_cheat == 1:
    print("Access denied!")
elif score >= 100 or is_premium == 1:
    print("Level unlocked")
else:
    print("Level locked")
```

13. Mobile Data Usage

```
data-used = float(input("Enter data used (GB): "))
unlimited-plan = input("Unlimited plan? (yes/no): ")
roaming = input("Roaming? (yes/no): ")
if (data-used <= 2 or unlimited-plan == "yes") and roaming == "no":
    print("Unlimited data available")
else:
    print("Data limited")
```

14. Office Entry System

```
id-valid = input("ID valid? (yes/no): ")
fingerprint = input("Fingerprint match? (yes/no): ")
face-scan = input("Face scan match? (yes/no): ")
holiday = input("Is it a holiday? (yes/no): ")
if id-valid == "yes" and (fingerprint == "yes" or face-scan == "yes")
    and holiday == "no":
    print("Entry allowed")
else:
    print("Entry denied")
```

15. Movie Rating Display

```
rating = float(input("Enter average rating: "))
editors_choice = input("Editor's choice? (yes/no): ")
if (editors_choice == "yes"):
    print("Recommended")
elif rating >= 8.5:
    print("Excellent")
elif rating >= 6.0:
    print("Good")
else:
    print("Average")
```