(1) Explain programming and python in detail.

• Definition and purpose of programming.

Definition:- Programming is the process of developing and writing a set of instructions using a programming language so that a computer can perform a specific tasks.

Purpose:-

⟹ To solve problems using computers

⟹ To perform tasks automatically.

⟹ To develop applications, software, websites and systems

⟹ To process data and make decisions.

⟹ Improve speed and accuracy (Computer don't make calculations mistakes like humans)

⟹ Process and analysis data (weather prediction business reports, data analysis.

• characteristics and Application of python.

Python has several important characteristics that make it popular and easy to use.

⟹ Simple and easy to learn:- It has clear and English like syntax. So beginners can understand it easily.

⟹ High level language:- Programmers do not need to manage memory or hardware details. Python handles them automatically

⟹ Interpreted language:- Python executes code line-by line which makes debugging easier.

⟹ object-oriented:- Python supports objects and classes and concepts like inheritance and polymorphism

⟹ Platform independent:- Python program can run on ~~in~~ windows, linux and Mac without modification.

⟹ Dynamically typed:- Data types are assigned automatically at runtime, no need to declare them.

⟹ Large standard library:- Python provides many built-in modules for tasks like math, files handling, ~~mene~~ networking.

⟹ Open Source and Free:- Python can be downloaded freely and its source code is open to ~~a~~ everyone.

• Applications:- Following are the Applications of python.

⟹ Websites:- Used to make websites and web applications

⟹ Data Analysis:- Used to study and understand large amount of data.

⟹ Machine learning and AI:- Used to make smart system like face recognition etc.

⟹ Automation:- Used to do repeated tasks automatically

⟹ Games:- Used to make simple games

⟹ Cyber Security:- Used to make software for computers.

⟹ Robotics and IoT:- Used in small devices and robots Python is used for websites, data analysis and robotics

• Types of Comments in python with syntax.

Python supports three types of comments

⟹ Single-line Comment:- It is used to write a short Comment in one line and ~~sho~~ starts with

the '#' symbol.

Syntax:- # this is a single line Comment

→ Multi-line comment:- It is used to write long comments in multiple lines and written using triple quotes ''' ''' or """ """

Syntax:-
```
'''this is a
Multi line Comment
in Python
'''
```

→ In-line comment:- Comment written at the end of a statement and used to explain a specific part of code.

Syntax:-
```
x=10  #assigning value 10 to x
print(x) # printing the value of x
```

• Importance of python in modern software development:

Python is very important in todays software world because of its simplicity, power and wide usage.

Some key points are:

→ Easy to learn:- Python is simple, so developers can write programs faster.

→ Saves time: less code is needed, so software is built quickly

→ Used in modern Technologies:- Python is widely used in AI, machine learning, and data science.

→ Many libraries Available:- Python has ready made tools for web apps, data, AI, games etc.

→ Works on All platforms: Python programs run on windows, linux and Mac.

→ Used by Big Companies: Companies like Google, Netflix, and NASA use python.

→ Large Community:- Many people use python, so help and support are easily available.

⟹ Works in many fields:- Python is used in webapps, mobiles apps(basic use), Game development, AI/ML, IOT (smart devices), cloud and DevOps.

(2) Describe Data type and operations in python with suitable examples.

• Built-in data types in python (Numeric, Sequence, Set, Mapping, Boolean)

Datatypes:- A data type tells what kind of value is stored in a variable, such as numbers, text, true/false etc.

Python has different data types.

⟹ Numeric Data types: Numeric data types are used to share numbers. They are three types of numeric data types.

• int - whole numbers (no decimals)
• float - Decimal numbers
• Complex Numbers with imaginary part (a+bj)

Example:- x = 100 #int
y = 3.14 #float
z = 2+3j #complex

⟹ Sequence Data types:- These store multiple items in an ordered manner.

ommon Sequence types are:

- String (str):- Stores text, string data types are written in Quotes

  name = "python"

- List:- these are ordered mutuable and allows duplicates list are written in "[ ]"

Example:- fruits = ["apple", "banana", "Mango"]

- Tuple:- these are ordered but not changeable. Tuples uses "( )"

Example:- Colors = ("red", "green", "blue")

⇒ Set data types: Stores data in key: Value pairs It uses "{ }". Only dictionary belongs to mapping type.

Example: numbers = {1, 2, 3, 4}

⇒ Mapping data types:- stores data in key: Value pairs. It uses "{ }". Only dictionary belongs to mapping types.

Example:- Student = {"name": "Akhila", "age": 20}

Here "name" and "age" are keys

"Akhila" and 20 are values

⇒ Boolean Data types: Stores True or False values. used in conditions and logic. Bookean often comes from Comparision.

Example: Print (10>5) # True

Print (2==3) # False

- Type Identification using type()

In python, you can identify the type of a variable or value using the built-in type() function. It tells you what kind of objects it is (like int, str, list, etc)

Syntax: type (object)

- Object → this variable or value you want to check.
- Returns the type of the object

Example:

- Numeric types:

```
x=10
y=3.14
Print(type(x)) #<class 'int'>
print(type(y)) #<class 'float'>
```

- string type

```
name = "Akhila"
print(type(name)) #class 'str'>
```

- Boolean type

```
is_valid = True
print(type(is_valid)) #<class 'bool'>
```

- List, Tuple, set, Dictionary

```
my_list = [1,2,3]
my_tuple = (1,2,3)
my_set = {1,2,3}
my_dict = {"a":1, "b":2}
Print(type(my_list)) #class 'list'>
Print(type(my_tuple)) #<class 'tuple'>
print(type(my_set)) #<class 'set'>
print(type(my_dict)) #<class 'dict'>
```

NOTE: You can also use "is instance()" if you want to check if a variable belongs to a certain type

Example: x=10

```
Print(is instance(x, int) # True
Print(isinstance(x, float) # False
```

Various Python Operators (Arithmetic, Assignment, comparsion, Logical, Membership, Identity)

Operator: An operator is a symbol that tells the computer to perform certain mathematical and logical operations These are different operators in python.

⇒ Arthemetic operations: these are used for doing some basic mathematical calculations.

Operator	Meaning	Example	Result
+	Addition	10+5	15
-	Subtraction	10-5	3
*	Multiplication	10*5	50
/	Division	10/5	2.0
//	Floor division	10//5	3
%	Modulus	10%3	1
**	Exponent (Power)	2**3	8

⇒ Assignment Operators: Used to assign values to variables

Operator	Meaning	Example	Equivalant
=	Assign	x = 5	x = 5
+=	Add & Assign	x += 3	x = x+3
-=	Substraction & Assign	x -= 3	x = x-3
*=	Multiply & Assign	x *= 3	x = x*3
/=	divide & Assign	x /= 3	x = x/3
//=	floor & Assign	x //= 3	x = x//3
%=	Modulus & Assign	x %= 3	x = x%3
**=	Power & Assign	x **= 3	x = x**3

⇒ Comparision (Relational operators): Used to Compare values. Returns True or False

Operators	Meaning	Example	Result
==	Equal to	5==5	True
!=	Not equal	5!=3	True
>	Greater than	5>3	True
<	less than	5<3	True
>=	Greater or Equal	5>=5	True
<=	less or Equal	3<=5	False

⇒ Logical operators: Used for logical decisions works with conditions.

operator	Meaning	Example	Result
and	True if both True	True & false	false
or	True if atleast one true	True & False	True
not	Reverse the condition	not True	False

⇒ Membership operators: Used to test if a value exists in a sequence (List, string, tuple, set)

Operator	Meaning	Example	Result
in	True if value Present	"a" in "apple"	True
not in	True if not present	"b" in "apple"	True

⇒ Identity operators: Used to compare memory locations of two objects.

operator	Meaning	Example	Result
is	True if same object	x is y	True/false
is not	True if not same object	x is not y	True/false

Explain python Input and output operations in detail.

• Input() function and its default data type.

→ Input() Function: Input() function are used to take input from the user.

Syntax: var = input("Message")

It always returns data as string by default.

⟹ Default data type: Data type taken using input() is always stored as str (string data type)

Type Conversion (casting)

To convert input to other data type:

⟹ Convert to integer:

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

⟹ Convert to float:

```
price = float(input("Enter Price:"))
```

⟹ Convert to boolean (manually interpreted)

```
Example: x = input("Enter anything:")
         print(type(x)) # output: <class 'str'>
```

Key points:

- Input() function pauses program & waits for user Input
- Useful for interactive programs
- Requires type conversion for mathematical operations
- Types Conversion while taking input.

⟹ input() always takes data as string (str) by default.

⟹ To use numbers (for math, we must convert the input

⟹ Common Conversions:

int() → Conversion Converts input to integers

float() → Converts input to decimal numbers

Example: x = [1,2,3]
y = x
z = [1,2,3]
print (x is y) # True (same object)
print (x is z) # False (same value, diff object)

- Real-world usage of operators

⟹ Arithmetic operations: Used for calculating things in daily life.

Example: Total bill, masks, salary, distance

⟹ Assignment operators: used to store or update value in a program.

Example: adding points, updating balance, counting items.

⟹ Comparision operators: used to compare values

Example: Checking if age is 18 or not
Checking if marks are above pass marks
Checking if stock is available

⟹ Logical operators: Used to combine conditions

Example: Login only if emails & Password are correct
Can vote if age > 18 AND citizen.

⟹ Membership operators: Used to check if something in a list or group exists.

Example: checking if a name is in Contacts
checking if an item is in a soshopping Cart

⟹ Identity operators: Used to check if two things are exactly same object in memory.

Example: checking if two variables refer to the same user session.

Online Exam Result

A student passes if:

theory ≥ 40 AND Pratical ≥ 40

But if total (theory + practical) ≥ 100, pass even if one is less than 40. Input: theory, practical.

```
theory = int(input("enter theory marks"))
practical = int(input("enter practical marks"))
if (theory >= 40 and practical ≥ 40) or (theory + Practical >= 100):
    Print("pass")
else:
    Print("Fail")
```

*Hotel Room Pricing

A hotel charges:

₹ 3000 per day for normal days

₹ 4000 per day on weekends

If customer stays more than 3 days, give 15% discount

Input: is weekend (1 or 0), days stayed

print final bill.

Code:-

```
isweekend = int(input("Enter 1 for a weekend, 0 for not a weekend: "))
number_days = int(input("Enter the days stayed: "))
rate = 4000 if isweekend == 1 else 3000
total_bill = rate * days stayed
if days stayed > 3:
    bill = total_bill * 0.15
    final_bill = total_bill - bill
```

```
else:
    final_bill = total_bill
Print("final Bill:", final_bill)
```

* Gaming level unlock
A game unlocks next level if:
Score ≥ 100
OR
Player has a premium pass
But if player used cheating, access is denied.
Input: Score, is premium, used cheat.
Code:

```
Score = int(input("enter the Score"))
Pre = int(input("enter 1 if it is premium, 0 if not"))
cheat = int(input("enter 1 if used cheat, 0 if not"))
if usedcheat == 1;
    Print("Access Denied")
elif score >= 100 or pre == 1.
    Print("Level unlocked")
else:
    Print("Locked")
```

* Student scholarship
A student gets a scholarship if:
marks ≥ 85
AND
family income < 500000
But if student is a single parent child, income condition is ignored. Input: marks, income, single parent (1 or 0)

Code:

```
marks = int(input("enter the marks"))
income = int(input("enter the income"))
Single parent = int(input("enter student is singleparent child"))
if marks >= 85 and(income < 50000 or single parent == 1):
  Print ("scholarship Granted")
else:
  Print ("Not Eligible")
```

* Electricity Bill

units consumed

First 100 units → ₹2/unit

Next 100 units → ₹3/unit

Above 200 units → ₹5/unit

Note: No loops

print final bill amound.

Code:

```
Units = int(input("enter the units consumed"))
 if units <= 100:
     bill = units * 2
elif units <= 200:
     bill = 100*2 + (units - 100)*3

else:
     bill = 100*2 + 100*3 + (unit-200)*5
Print ("Total Bill :", bill)
```

* Bank loan Approval.

A bank approves a loan if:

salary ≥ 30000 AND credit score ≥ 700

OR

Salary ≥ 50000 (credit score ignored)
Input: salary; Credit score
Print "Loan Approved" or "Loan Rejected"
Code:-

```
Salary = int(input("enter the Salary"))
cs = int(input("enter the credit score"))
if salary >= 50000 or (salary >= 30000 and creditscore >= 700):
    print("Loan Approved")
else:
    print("Loan Rejected")
```

* Online food Delivery
A Restaurant gives free delivery if: order amount ≥ 500 OR user is a gold member
But if the distance, is Gold (1 or 0), distance. free. Input: amount, is Gold (1 or 0), distance. is more than 10km, delivery is never.
print "Free Delivery" or "Delivery charged".
Code

```
amount = int(input("enter the amount"))
is Golal = int(input("enter user is a gold member"))
distance = int(input("enter the distance"))
if distance > 10:
    print("Delivery charged")
elif amount >= 500 or Gold == 1:
    print("Free Delivery")
else:
    print("Delivery charged")
```

13) Program on Mobile Data usage.

```
data used = int(input("Enter data used in GB"))
unlimpla = int(input("Enter 1 if you have unlimited plan
                      else 0"))
is Roaming = int(input("Enter 1 if roaming is on else 0"))
if (data used <= 2 or unlimpla == 1) and is Roaming == 0:
  Print("A network gives unlimited data")
elif:
   Print("Limited data applies")
elif is Roaming == 1:
   Print("unlimited plan doesnot work")
```

* Output:

```
Enter data used in GB: 5
Enter 1 if you have unlimited plan else 0: 0
Enter 1 if roaming is on else 0: 0
Limited data applies
```

14) Program on office Entry system.

```
idvalid = int(input("Enter 1 if id is valid else 0: "))
Finpri = int(input("Enter 1 if fingerprint is valid else 0:"))
facescan = int(input("Enter 1 if facescan is valid else 0: "))
is holiday = int(input("Enter 1 if holiday else 0: "))
if idvalid == 1 and (finpri == 1 or facescan == 1):
  if isholiday == 1:
    Print("Access denied today is Holiday")
else:
   Print("Enter into office")
else:
   Print("Access denied")
```

* Output:

Enter 1 if id is valid else 0: 1
Enter 1 if finger print is valid else 0: 1
Enter 1 if facescan is valid else 0: 0
Enter 1 if holiday else 0: 0
Enter into office

15) Program on Movie Rating Display.

```
Average = float(input("Enter rating"))
isedicho = int(input("Enter 1 if it is editor choice else 0:"))
if isedicho == 1:
    Print("recommended")
elif average >= 8.5
    Print("Execellent")
elif average >= 6.0
    Print("Good")
elif else:
    Print("Average")
```