1. Explain programming and python in detail

• Definition and purpose of programming

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

Purpose :

* To solve problem using computers.
* To perform tasks automatically.
* To develop application software and systems.
* To process data and make decission.
* Improve speed and accuracy (computers don't make calculation mistake like humans).
* Process and analyze data (weather prediction, data analysis)

• Characteristics and application of Python.

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

* Simple and easy to learn : Python has clear and English-like syntax, so begginers 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 classes and objects and concepts like inheritance and polymorphism.
* Platform Independent : Python program can run on windows, linux and Mac without modification

* Dynamically typed : Datatypes are assigned automatically at run time, no need to declare them.
* Large Standard library : Python provides many built-in modules for tasks like math, file handling, networking etc

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 computer.
* Robotics and IOT : Used in small devices and robots

python is used for websites, data analysis, AI/ML automation games, apps, security and robotics.

Type of Comments in python with Syntax.
Python supports three types of Comments.
* Single line Comment : Used to write a short Comment in one line and starts with # symbol.

Syntax : # This is a single line Comment.
* Multi-line Comments : Used to write long comment in multiple lines and written using triple quotes.
''' ''' or """ """

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

5. 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 to x
print(x) # printing value of x.

Importance of python in modern software development.

Python is very important in today's 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.
* Works on All platforms : Python programs run on Windows, linux, and Mac.
* Large Community : many people use python, so help and support are easily available.
* Works in many fields : python is used in web apps, Mobile apps (basic use), Game development, AI/ML, IOT (smart devices) cloud and Devops.

2. Describe Data Types and Operators in python with suitable examples.

• Built-in datatypes in python (Numeric, Sequence, Set, Mapping, Boolean).

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

Python has different data types.

* Numeric Datatypes : Numeric datatypes are used to store numbers. There are three types of numeric datatypes.
  - int - whole numbers.
  - float - Decimal numbers
  - Complex Numbers with imaginary part (a+bj)

Ex : x = 10 #int
y = 3.14 #float.
z = 2+3j # complex.

* Sequence Datatype : These store multiple items in an ordered manner.

Common Sequence types are.

- String (str) : Stores text, string datatypes are written in Quotes.
  name = "python".
- list : These are ordered, mutable and allows duplicates list are written in "[ ]".
  Ex : fruits = ["apple", "banana", "kiwi"]
- Tuple : These are ordered but not changeable. Tupiles uses "( )"
  Ex : colors = ("red", "orange", "blue")

* Set Datatypes : Stores unorderded data and no duplicates are allowed. Set Datatypes uses "{ }"

Mappin Data types : Stores data in key : value pairs. It uses "{}". Only dictionary belongs to mapping type.

Ex : Student = {"name" : "Manu", "age" : 21}

Here, "name" and "age" are keys.

"Manu" and "21" are values.

- Boolean Datatypes : Stores True or False values. used in conditions and logic Boolean often comes from the comparisions.

Ex : print (10 > 5) # True.

print (2 == 3) #False

Type Identification using type()

In python you can identify the type of variable or value using the built-in type() function. It tells you what kind of object it is

Syntax : type (object)

Object : The variable or value you want to check.

- Returns the type of the object.

Ex :

- Numeric types.

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

- String type.

```
name = "Raju"
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 "isinstance()" if you want to check if a variable belongs to certain type.

Ex: x = 10

Print(isinstance(x, int)) #True

Print(isinstance(x, float)) #False.

• Real-World usage of Operators.

* Arithmetic Operators: Used for calculating things in daily life.

Ex: Total bill, marks, Salary, distance.

* Assignment operators: Used to store or update values is a program.

Ex: adding points, updating balance, Counting items.

* Comparission Operators: Used to compare values.

Ex: Checking if age is 18 or not

• Checking if marks are above pass marks.

* Logical Operators: Used to combine conditions.

Ex: ~~Used~~ to login only if email & password are correct.

• Can vote if age > 18 and citizen.

* Membership Operators: Used to check if something in a list or group exists.

: • Checking if a name is in contacts.
• Checking if an item is in a shopping cart.

* Identity Operation : Used to check if two things are exactly same object in memory
Ex : Checking if two variables refer to the same user session.

5. Explain python Input and Output Operators in detail.
• Input() function and its default datatype
* Input() Functions : Input() function are used to take input from the user.
Syntax : var = input("Message")
~~It always stored as str (str~~
It always returns data as string by default.
* Default datatype : Data takes using input() is always stored as str (string type).
Types Conversion (casting)
To Convert input to other datatypes :
⇒ Convert to integer :
num = int(input("Enter a number"))
⇒ Convert to boolean
Ex : x = input("Enter anything :")
Print(type(x)) # Output : <class 'str'>
⇒ Convert to float.
price = float(input("Enter price :"))

Key points :
• Input() function pauses program & waits for user Input
• Useful for interactive programs.
• Requires type Conversion for mathematical Operations.

Type Conversion while taking input.

* input() always takes data as string (str) by default. We must convert it
* To use numbers (for math), we must convert the input.

* Common Conversions:
  - int() → Converts input to integer.
  - float() → Converts input to decimal number.

Ex: a = int(input("Enter age:"))
    b = float(input("Enter price:"))

* Without Conversion:
  - x = input("5") → "5" (string)
  - y = input("3") → "3" (string)
  - x+y = "53" (string concatenation)

* With Conversion:
  - x = int(input("5")) → 5 (integer)
  - y = int(input("5.8")) → 5.8 (decimal)

* input() → str.
  - int(input()) → integer
  - float(input()) → decimal number

Type Conversion is needed for arithmetic Operation.

Taking Multiple inputs:

Some times we need more than one input from the user.

Method 1:

a, b = input().split()

=> User enters value in one line
→ Default type is string

Ex: a, b = input("Enter two numbers:").split()

Type Conversion with split():

If we need integer values:

```
a, b = map(int, input().split())
```

If we need float values:

```
x, y = map(int, input().split())
```

Ex:
```
a, b = map(int, input("Enter two numbers: ")).split()
print(a+b)
```

Here, split() → breaks input by space.
map() → Converts each value.

- Formatted Output using print(), Seperators and format specifiers

* Print() is used to display output.
* Sep - changes the seperator between values.
  Ex: print(1,2,3, Sep = " = ") # 1 = 2 = 3
* "End" changes the ending of the line
  Ex: Print("Hello", end = " ")
* Format is used to insert values in a formatted way.
  Ex: "Name = {}".format(name)
* f - Strings are used for modern formating
  Ex: f "Name = {name}"
* format specifiers:
  %.d → integer, %.f → float, %.s → string, %.2f → 2 decimal places.

4. Discuss Control Statements and Decission Making Statements in Python.

- Meaning and importance of Control Statements.

Meaning: Control statements are special instructions in a programming language that control the flow of execution of a program. They decide which part of the code runs, how

many times it runs, and under what condition it runs

Importance of Control Statements.

* Decision Making: Control statements allow program to make decisions and choose different actions based on conditions
* Repetition of tasks: Loops help execute a block of code multiple times without writing it repeatedly.
* Better program flow & control: They allow developers to control the order in which instruction on executed improving efficiency.
* Enhances program logic: They make it possible to implement real-life logic in software.

* Types of Control Statements:

In python Control statements are mainly of three types They are

* Conditional (Decission-making statements): Used to check conditions and decide what to do
  - if
  - if-else
  - elif
* Looping (Repetition) Statements: Used to repeat a block of code multiple lines.
  - for
  - while.
* Jump Statements: Used to change the normal flow inside loops.
  - break → stops the loop
  - Continue → skips one iteration and continues
  - return → exist a function.

*) Decission-making Statements: if, if-else & if-elif-else

→ if statement :

- Used when you want to check one condition.
- If the condition is true, then then code run.

Ex : age = 18
```
if age >= 18 :
    Print ("you are an adult")
```

→ if-else statement :

- Used when you have two possible outcomes.
- If the condition is true, one block runs.
- If the condition is false, another block runs.

Ex : marks = 30.
```
if marks >=35 :
    Print("Pass")
else :
    print("Fail")
```

→ if-elif-else statement :

- Used when you have multiple conditions
- elif means else if.
- It check conditions one by one.
- Only the first true condition runs.

Ex : marks = 75
```
if marks >=90 :
    print ("Grade A")
elif marks >=75 :
    print("Grade B")
elif marks >=60 :
    Print("Grade C")
else
    print ("Grade D")
```

- Syntax flow and execution control with example.

~~if statement~~

5. Write an essay on python programming fund

• Role of programming in problem Solving:

Programming plays an important role in solving proble because it helps us tell the computer what to do in a clear, and step-by-step way.

* Breaking problem into steps : Big problems can be divided into smaller and easier steps, so it becomes simpler to solve.
* Doing tasks automatically : Programs make the computer do work by itself without repeating the same work manually.
* Giving fast and Correct results : Computer follow instructions exactly, so the results are faster and more accurate than humans.
* Handling big amounts of information : Programming allows the computer to store, manage and process large data that humans cannot

• Python syntax Simplicity and readability.

Python is known for having a simple and clean syntax. This means the way we write python code is

* easy to understand
* easy to write.
* close to normal English language.

⇒ Simplicity in Syntax : Python focuses on writing less code with clear meaning. It avoids complicated symbols and unnecessary rules.

For example, In python we don't need ':' or '{}' for basic

Structure:

Indentation (spaces) is used to show blocks of code which makes it simple.

Readability: Readability means the code is easy to read and understand even after many months. Python code looks clean and neat, like plain English instructions.

Because of this.

- develops can find errors easily.
- teams can work together better.
- Programs are easier to maintain.

Example: if age > 18:
    print ("Adult")

This reads like a simple english sentence.

- Use of comments for code documentation.

Comments are notes written inside a program that are not executed by the computer. They are used only for explaining the code to humans.

Why comments are useful.

- Explain what code does: Comments help others understand the purpose of code.
- Make code easier to read: They describe logic in simple words, which improves readability.
- Help in Debugging: By commenting out certain lines temporarily, we can test ~~ordinarry~~ or debug the program.

Here, The second comment explains the line.
The first comment explains the function.

1. Program on Movie Ticket Pricing

```
age = int (input ("Enter your age:"))
is3D = int (input ("Enter 1 if you are watching 3D
        movie else enter 0:"))

if is3D == 1
    if age < 13:
        Print ("your ticket price is ₹200")
    elif age >=13 and age<=59:
        Print ("your ticket price is ₹300")
    else:
        Print ("your ticket price is ₹250")
elif age<13:
    Print ("your ticket price is ₹150")
elif age >=13 and age <=59.
    Print ("your ticket price is ₹250")
else:
    Print ("your ticket price is ₹200")
```

Output:

Enter your age: 11

Enter 1 if you are whatching a 3D movie else enter 0: 1

Your ticket price is ₹200

gram on College Attendance Rule:

```
attper= int (input ("Enter your attendance percentage :")
medcer =int(input("Enter 1 if you have medical certificate
                                    else 0 :"))
if attper >=75 or (attper >=60 and medcer ==1):
    Print ("You are ~~not~~ allowed to write exam")
else:
    Print ("You are not allowed to write exam")
```

Output:

Enter your attendance percentage :20
Enter 1 if you have medical certificate else 0 :1
You are not allowed to write exam.

3. Program on E-Commerce Discount.

```
billamo = int(input("Enter your bill amount :"))
isPrime = int(input("Enter 1 if you are prime member else 0:"))
if isPrime==1:
    if billamo>=5000:
        dis =0.25*billamo
    elif 2000<=billamo<=4999:
        dis = 0.15 * billamo
    else:
        dis=0
else
    if billamo>=5000:
        dis =0.20 * billamo
    elif 2000<=billamo<=4999:
        dis = 0.10* billamo
    else:
        dis =0
amo = billamo-dis
Print("Final amount to be paid is", amo)
```

Output :

Enter your bill amount : 4500

Enter 1 if you are prime member else 0 : 0

Final amount to be paid is 4050.0

4. Program on Smart phone battery Warning

```
batper = int(input("Enter your battery percentage"))
ischar = int(input("Enter 1 if phone is charging else 0"))
if ischar == 0:
    if batper <= 20:
        print("Low battery")
    elif 21 <= batper <= 80:
        print("Normal")
    else:
        print("full")
elif ischar == 1:
    print("Charging")
else:
    print("Enter 0 or 1")
```

Output :

Enter your battery percentage : 81

Enter 1 if battery is charging else 0 : 1

Charging

gram on Driving license check.

```
age = int (input ("Enter your Age:"))
testpassed = int (input ("Enter 1 if you passed the test
                          else 0:"))
if (age >=18 and testpassed == 1) or age >= 60:
    Print ("Eligible")
else:
    Print ("Not Eligible")
```

Output:

Enter your Age: 60

Enter 1 if you passed the test else 0: 1

Eligible

6. Program on Online food delivery

```
amount = int (input ("Enter Order amount"))
isGold = int (input ("Enter 1 if you are a Gold member
                      else 0"))
dis = int (input ("Enter distance in km:"))
  if (amount >= 500 and dis < 10) or (isGold == 1 and dis<40):
    Print ("Free delivery")
  else:
    Print ("delivery is never free").
```

Output:

Enter order amount: 700

Enter 1 if you are a Gold member else 0: 1

Enter distance in km: 11

delivery is never free.

7. Program on Bank loan Approval

```
Sal = int(input("Enter your Salary :"))
Cresco = int(input("Enter your credit score :"))
if (sal >= 3000 and cresco >= 100) or Sal > 50000:
    Print("Loan Approved")
else:
    Print("Loan Rejected.
```

Output :

Enter your Salary : 70000
Enter your credit score : 800
loan Approved.

8. Program on Electricity Bill

```
units = int(input("Enter number of units consumed :"))
if units <= 100:
    bill = units * 2
elif units <= 200:
    bill = 100 * 2 + 100 * 3 + (units - 200) * 5
Print("Final bill amount : ₹ ", bill)
```

Output :

Enter number of units consumed : 250
Final bill amount : ₹750

9. Program on Student Scholarship.

```
marks = int(input("Enter your marks:"))
famInc = int(input("Enter family income"))
isSimpar = int(input("Enter 1 if you have single Parent
                      else 0 :"))
if issimpar == 1 and mark >= 80:
```

```
Print ("You are eligible")
if marks >= 85 and famInc < 500000:
    Print ("You are eligible")
else:
    Print ("You are not eligible")
```

Output:

```
Enter your marks: 85
Enter family income: 60000
Enter 1 if you have single parent else 0: 0
You are eligible
```

10 Program on online Exam Result

```
theory = int(input("Enter your theory marks"))
practical = int(input("Enter your practical marks"))
total = theory + practical
if total >= 100 or (theory > 40 and practical >= 40):
    Print ("Pass")
else:
    Print("Fail")
```

Output:

```
Enter your theory marks: 50
Enter your practical marks: 50
Pass
```

11. Program on Gaming Level Unlock.

```
Score = int(input("Enter your game score:"))
ispre = int(input("Enter 1 if you have premium or else 0:"))
usedcht = int(input("Enter 1 if you used cheating else 0:"))
if (Score >= 100 or ispre == 1) and usedcht == 0:
    Print("Next Level")
```

```
elif usedcht == 1:
    Print ("Access denied")
else :
    Print ("Next level is locked")
```

Output :

Enter your game score : 60
Enter 1 if you have premium else 0 : 0
Enter 1 if you used cheating else 0 : 0
Next Level.

12. 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 ("Excellent")
elif average >= 60 and average <= 8.4 :
    Print ("Good")
else :
    Print ("Average")
```

Output :
Enter rating : 8.5
Enter 1 if it is editor choice else 0 : 0
Excellent