M. Manoranjani
III rd Biotechnology
Admin No: 20055

Explain programming and python in detail?

Programming is the process of writing instructions that tells a computer what to do. These instructions help the computer solve problems or perform tasks like calculations, data storage / decision making.

Exaample:
A program to add two numbers

```
a = 10
b = 20
Print (a+b)
```

What is Python?

Python is a high-level programming language that is easy to learn and understand. It was developed by "Guido Van Rossum" and is widely used because of its simple syntax and readability.

Characteristics of Python
→ Easy to read and write
→ elses simple english-like syntax
→ Large collection of libraries
→ Free and open-source
→ supports multiple programming styles.

Applications of python
→ Web development
→ Data analysis
→ Automation
→ scientific research

Example:-

```
print ("Hello, World")
```

Types of comments in Python

Comments are statements in a program that are not extended by Python. They are used to explain the codes and improve readability.

1. Single line Comments
⇒ Used to explain a single line of code
⇒ Begins with the # symbol.
⇒ Python ignores this line during execution.

Syntax:
```
#This is a single-line comment
```
Example:
```
# Assign value to variable
x = 10
```
2. Multi-line Comments
→ Used to explain multiple lines at once
→ Written using triple quotes (''' or """)
→ Commonly used for documentation.

Syntax
```
''' '''
This is a
multi-line comment
''' '''
```
Example.
```
"""
This program calculates sum of two numbers
"""
a = 5
b = 6
print(a+b)
```

Importance of comments
→ Makes code easy to understand
→ Helps in debugging
→ Useful for documentation

Importance of Python in Modern Software Development
→ It is easy to learn and use
→ simple syntax reduces coding time
→ supports multiple programming styles
→ Has many libraries and frame works
→ used in AI, data science, web development and automation
→ Works on different operating systems.

Data Types and operators in Python.
Python provides different built-in data types to store various kinds of data and operators to perform operations on them.

Built-in Data Types in Python
1. Numeric Data types
used to store numbers such as integers and decimal values
```
a = 10      # integer
b = 3.5     # float
```

2. Sequence Data Types:

Tuple - ordered collection; immutable

Used to store a collection of values

```
name = "Manu" # string
marks = [80, 90, 85] # list - ordered collection; Mutable
```

3. Set Data Type

Stores unique elements

```
nums = {1, 2, 3}
```

List - [ ]
Tuple - ( )
Set - { }

4. Mapping Data Type

Stores data in key-value pairs

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

5. Boolean Data Type

Stores True or False Values

```
result = True
```

int = a = 10, b = -12, c = 1 2 3 4 5 6 7

float = x = 1.0 ; y = 12.3 ; z = 13.4

Complex = Alphabet with number ; A = 2 + 5j
(2 → real number; 5j → imaginary)

String - Sequence of characters represented in quotation marks.
In python we can use single, double / triple quotes to define a string

"hello Python"

Type Identification using type( )

The type () function in python is used to identify the data type of a variable. It tells what kind of values is stored in the variable.

```
type(variable_name)
a = 10
print(type(a)) # int
b = 3.5
print(type(b)) # float
name = "Python"
print(type(name)) # str
nums = [1, 2, 3]
print(type(nums)) # list
flag = True
print(type(flag)) # bool
```

Operators in Python (x + y) → operands, → operators

→ Used to perform operations on variables & values
→ Python operators are special symbols used to perform specific operations on one or more operands.

Unary Operators - Python operators that require one operand to perform a specific operation. known as unary operators.
Binary Operators - Python operators that require two operands to perform a specific operation.
Operands - Variables, values or expressions that are used with operator to perform a specific operation.

Types of Python Operators

Python operators are categorized in the following categories.

1. Arithmetic Operators

Used for mathematical calculations

Operator	Meaning	Example
+	Addition	10 + 5 = 15
−	Substraction	10 − 5 = 5
*	Multiplication	10 * 5 = 50
/	Division	10 / 5 = 2.0
%	Modulus (remainder)	10 % 3 = 1
//	Floor Division	10 // 3 = 3
**	Exponent	2 ** 3 = 8

Real-world use:-

calculating total marks, salary, discounts, average score etc.

2. Assignment Operators

Used to assign values to variables

Operator	Example	Meaning
=	x = 10	Assign value
+=	x += 5	x = x + 5
-=	x -= 5	x = x - 5
*=	x *= 2	x = x * 2

Real world use:-
updating bank balance, increasing counters, modifying scores.

3. Comparision Operators

Used to compare two values. Result is True / False

Operator	Meaning
= =	Equal
! =	Not equal
>	Greater than
<	Less than
>=	Greater than or equal
<=	Less than or equal

Real-world use.

checking pass/fail marks, age eligibility, login validation

4. Logical operators

Used to combine conditions

operator	Meaning
and	True if both conditions are true
or	True if any one condition is true
not	Reverse the result

Real-world use:

checking multiple conditions like age and qualification

Membership operators

used to check whether a value exists in a sequence

Operator	Meaning
in	Value exists
not in	Value does not exist

Example - 'a' in apple # True

Real-world use:

checking ~~object identity in programs~~ username in database, item in list

Identity Operators - used to compare memory locations

is	Same object
is not	Different object

Real world use.

checking object identity in programs

Python Input and Output Operator

Input operation in python

input ( ) Function

→ Used to take input from the user

→ Default data type is string (str)

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

Type conversion while Taking Input

Since input is always string, type conversion is required

```
age = int (input ("Enter age: "))
Salary = float (input ("Enter Salary: "))
```

Taking Multiple Inputs

Multiple inputs can be taken in one line using split ( )

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

Output operation in Python

print ( ) Function

used to display output on the screen.

```
print ("Hello Python")
```

Separator (sep)

used to separate multiple values

```
print (10, 20, 30, sep = ",")
```

output

10, 20, 30

Format Specifiers / Formatted Output

Using format ( )

```
print ("My age is {}".format (age))
```

using f - string

```
print (f" My salary is {salary}")
```

Real - world use

Displaying suports, student details, invoices, bills

Control Statements and Decision-Making statements:

Meaning of control statements
They control the flow of execution of a program.
They decide which statement to execute and when

Importance
→ Enables decision making
→ Makes programs logical and dynamic
→ Avoids unneccessary execution.

Types of central statements
1. Decision-making statements
2. Looping statements
3. Jump statements.

Decision-Making statements

1. If-statement

Executes a block of code only if the condition is true.

Syntax.

```
if condition:
    statement
```

Example

```
if makes >= 35:
    Print ("Pass")
```

2. If-else statement

Executes one block if condition is true, otherwise executes another block.

syntax:-

```
if condition:
    statement 1
else
    statement 2
```

Example

```
if age >= 18
    print ("Eligible to vote")
else
    print ("Not eligible")
```

3. If - elif - else statement

used to check multiple conditions

syntax

```
if condition 1:
    statement 1
elif
    condition 2:
    statement 2
else
    statement 3
```

Example

```
if marks >=75:
    print ("Distinction")
elif marks >=60
    print ("First class")
elif marks >=35
    print ("pass")
else :
    print ("Fail")
```

Execution Flow

→ Conditions are checked top to bottom

→ First true condition is executed

→ Remaining conditions are skipped.

Example: (If - else execution flow)

1. Condition age >=18 is checked

2. condition is False

3. else block executes

4. program continues after if - else.

Python executes only the block whose condition is true

indentation plays a crucial role in controlling flow

Write an Essay on Python programming fundamentals.

1. Role of programming in problem solving.

→ programming helps in solving real-world problems using logical steps.

→ It breaks complex problems into smaller, managable tasks
→ Enables automation of repetitive work
→ Helps in accurate calculations and data processing
→ Improves efficiency and saves time

2. Python Syntax simplicity and Readability

→ Python has simple and English-like syntax
→ Easy to learn for beginners.
→ Does not use braces { } or semi colons ;
→ Uses indentation to define code blocks
→ Programs are easy to read, write and maintain

3. Use of Comments for code Documentation

→ Comments explain the purpose of code statements
→ Improve readability and understanding of programs.
→ Helpful for debugging and future modifications
→ Single-line comments use #
→ Multi-line comments use triple quotes (''' or """)
→ Comments are not executed by the interpreter

4. Data Types, Operators and Input/output operations

→ Python supports built-in data type like "int, float, str and bool"
→ operators perform arithmetic, comparision, logical & assignment operations
→ input() function is used to take input from the user.
→ Default data type of input() is "string"
→ print() function is used to display output.
→ Supports formatted output for better presentation.

5. Control flow using Decision-making statements

→ control flow decides the order of execution in a program
→ Decision-making statements help in making logical choices.
→ If statement executes code when condition is true.
→ If else provides two way decision making
→ If elif-else handles multiple conditions
→ Makes programs flexible and dynamics

Conclusion:

→Python fundamentals form the base for advanced programming

→Simple syntax and powerful features make Python popular.

→Widely used in education, science, data analysis and software development.

① Movie Ticket Pricing

A movie threatre charges:

₹150 for children (age <13)

₹250 for adults (age 13-59)

₹200 for seniors (age ≥60)

If the person is watching a 3D movie, add ₹50 extra.

write a program that takes age and is 3D (1 or 0)

```
age = int(input("Enter age: "))
is 3D = int(input("Is it a 3D movie? (1 for yes, 0 for No): "))

if age <13:
    price =150
elif age <=59:
    price =250
else
    price =200

if is 3D==1
    price += 50
print("Final Ticket price: ", price)
```

② College Attendance Rule: A student is allowed to write the exam if:

attendance ≥75

OR

attendance ≥60 and has medical certificate (1=yes, 0=no).

Take attendence percentage and medical certificates as impact and print "Allowed or "Not Allowed".

```
attendence = int(input("Enter attendence percentage: "))
medical = int(input("Medical certificat (1 for Yes, 0 for No): "))
```

```
If attendance >= 75 or (attendance >=60 and medical == 1):
    print ("Allowed")
else:
    print ("Not Allowed")
```

5) E-commerce Discount

A shopping site gives

20%. discount if bill ≥5000

10%. discount if bill between 2000 and 4999

No discount if bill < 2000

But if the customer is a prime number they get extra 5%. discount

Input : bill amount ; is prime (1 or 0)

Print final amount to be paid.

```
bill = float (input ("Enter bill amount : "))
if bill >= 5000:
    discount = 0.20
elif bill >0.10
    discount =0.10
else
    discount =0.0
is prime = int (input ("Is the customer a prime member ? (1
            for Yes, 0 for No): "))
if is prime == 1:
    discount += 0.05
final amount = bill - (bill * discount)
print ("Final amount to be paid : ₹", final_amount)
```

Smartphone Battery Warning

A phone shows:

"Low Battery" If battery ≤20

"Normal" if battery between 21-80

"Full" : if battery >80

But if phone is charging, it should show "Charging" instead of any messsage.

Input : battery percentage, is charging (1 or 0)

```
battery = int(input("Enter battery percentage : "))
if charging = int(input("Is the device charging "(1 for yes, 0 for No): "))

if Ischarging == 1:
    Print ("charging")
else
    If battery <=20:
    Print ("Low Battery")
```

Driving License check.

A person can get a driving license if:

age ≥18

AND

Passed driving test (1 =yes)

But if age ≥60, driving test is not required.

Input age, test passed.

Print "Eligible or "Not eligible"

```
age = int(input("Enter age: "))
test passed = int(input("passed driving test?(1 for yes, 0 for No): "))

if  age >=60 :
    Print ("Eligible")
elif age >=18 and test passed == 1:
    print ("Eligible")
else :
    print ("Not eligible")
```

Online Food Delivery

A restaurant gives free delivery if:

order amount ≥500

OR

user is a gold member

But if the distance is more than 10km, delivery is never free.

Input: amount, is Gold (1 or 0), distance

```
amount = float(input("Enter order amount : "))
is Gold = int(input("Is Gold member ? (1 for Yes; 0 for No): "))
distance = float(input("Enter delivery distance (km): "))
```

```
if distance >10:
    print ("Delivery charged")
elif amount >= 500 or is Gold == 1:
    print ("Free Delivery")
else:
    print ("Delivery charged")
```

* Bank loan Approval

A bank approves a loan if:
Salary ≥ 30,000 AND credit score ≥ 700
Input: Salary, credit score.
Print "Loan Approved" or "Loan Rejected".

```
Salary = int(input("Enter salary: "))
credit score = int(input("Enter credict score: "))
if Salary >= 50000 or (salary >= 30000 and creditscore >= 700):
else:
    print ("loan Rejected")
```

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 amount

```
units = int(input("Enter units consumed: "))
if units <=100:
    bill = units * 2
elif units <=200:
    bill = (100*2) + (units - 100) * 3
else:
    bill = (100*2) + (100*3) + (units - 200) * 5
print ("Electricity Bill Amount : ₹", bill)
```

Student Scholorship

A student gets a scholorship if:
marks ≥ 85
AND
family income < 500000

But if the student is a single parent, child, income condition is ignored

Input: marks, income, single parent (1 or 0)

```
marks = int(input("Enter marks: "))
Income = int(input("Enter family income: "))
Single parent = int(input("single parent child? (1 for yes, 0 for No): "))
if marks >=85 and (income < 500000 or single parent == 1):
    print("Scholarship Greanted")
else:
    print("Scholarship Not Granted")
```

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: Isweekend (1 or 0), days stayed

print final bill.

```
isWeekend = int(input("Is it a weekend stay? (1 for yes, 0 for No): "))
days stayed = int(input("Enter number of days stayed: "))
if Is weekend == 1:
    rate = 4000
else:
    rate = 3000
bill = rate * daysstayed
if days stayed > 3:
    bill = bill * 0.85
print("Total bill amount: ₹", bill)
```

Gaming level unlock

A game unlocks next level if:

Score ≥100

OR

Player has a premium pass

But if player used cheating acess is denied

input: score; is premium, used cheat

```
Score = int(input("Is premium player? (1 for yes, 0 for No): "))
if used_cheat == 1:
    print("Access Denied")
elif score >= 100 or is_premium == 1:
    print("Next Level unlocked")
else
    print("Level Locked")
```

3 Mobile Data Usage

A network gives unlimited data if:
daily usage < 2GB:
OR
user has unlimited plan
But if roaming is on, unlimited plan doesnot work
input : data used, has unlimited plan, is Roaming

```
data_used = float(input("Enter daily data usage (GB): "))
has_unlimited_plan = int(input("Has unlimited plan? (1 for yes, 0 for No): "))
is_Roaming = int(input("Is roaming on? (1 for yes, 0 for No): "))
if is_Roaming == 1:
    if data_used <= 2:
        print("unlimited data")
    else
        print("Limited Data")
else:
    if data_used <= 2 or has_unlimited_plan == 1:
        print("unlimited Data")
    else:
        print("Limited Data")
```

Office entry system

An employee can enter the office if
ID card is valid
AND
.fingerprint matches OR face scan matches)
But if it is a holiday, entry is denied for everyone.
input idvalid, fingerprint, face scan, is Holiday

```
Idvalid = int(input("For scan match ? (1 for yes, 0 for No): "))
Fingerprint = int(input("Finger print match ? (1 for yes, 0 for No): "))
facescan = int(input("Face scan match ? (1 for yes, 0 for No): "))
isHoliday = int(input("Inpolliday a holiday ? (1 forgyes, 0 for No): "))
if isHoliday == 1:
    print("Entry Denined")
elif Id valid == 1 and (finger print == 1 or face scan == 1):
    print("Entry Allowed")
else:
    print("Entry Denied")
```

Movie Rating Display

A movie app shows rating based on ~~average~~ average score

Average ≥ 8.5 → "Excellent".

Average between 6.0 and 8.4 = "Good"

Average < 6.0 = "Average"

But if the movie is marked as editor's choice always show "Recomended"

input: overage Rating, is editors choice (1 or 0)

Print the message.

```
average Rating = float(input("Enter average rating: "))
is editors choice = int(input("Is Editor's choice ? (1 for yes,
0 for No): "))
if is Editors choice == 1:
    print("Recommended")
elif average Rating >= 8.5:
    print("Excellent")
elif average Rating == 6.0:
    print("Good")
else:
    print("Average")
```