1. Explain programming and Python in detail.

Definition and purpose of programming

Programming is the process of writing instructions that tell a computer what to do. These instructions are written using programming languages.

Purpose of programming:

→ Solve problems
→ Automate tasks
→ Process data
→ Build applications (web, mobile, desktop)
→ Control machines and devices

Example: `print("Hello, world!")`

This program tells the computer to display a message.

Characteristics and Applications of Python

Characteristics:

→ Simple and easy to learn
→ High-level Language
→ Interpreted Language
→ Platform Independent
→ Large Library Support
→ Open source

Applications:

→ Web development (Django, Flask)
→ Data Science and Artificial intelligence
→ Game Development
→ Automation and Scripting
→ Mobile and desktop apps.

Types of comments in Phython with syntax

Single line comment: #This is a comment

```
print ("phython")
```

*multi line comment:

```
""" This is multi-line comment used for documentation"""
print("Hello")
```

Importance of phython in Modern software development

→ Faster development

→ Used by Google, NASA, Instagram

→ Supports AI, Machine learning, Data Analytics

→ Easy integration with other Languages

→ Strong community support.

2. Describe data types and operators in Phython with suitable Examples

Built-in data types in phython (Numeric, Sequence, Set, mapping, Boolean)

Numeric: int, float, complex (Stores whole numbers, positive(or) negative) Int:

Example: a = 10 (int)
b = 3.5 (float) → (Stores decimal numbers)
c = 3+5i (complex) → (Stores numbers in a+bj format)

Sequence: list, tuple, string.

name = "phython" (str) → (Stores text (or) characters), written inside quotes

numbers = [1, 2, 3] (list) → (Ordered collection, mutable)

colors = ("red", "blue", "green") → (Ordered collection, Immutable)

Set:

Example: s = {1, 2, 3} Set: (unordered collection, No duplicate values)

Mapping (Dictionary): (Stores data is Key-Value pairs)

Example: Student = {"name": "Ram", "age": 20}

Bookan: Returns True (or) false, used in decision making

Example: is pass = True.

Type identification using type()

Example: x = 10

print (type (x)) # <class 'int'>

Various phython operators:

(i) Arithmetic operators: Arithmetic operators are used to perform mathematical calculations.

→ They work with numbers like integers and floats

→ Used for mathematical calculations

| Operator | Meaning |
|---|---|
| + | addition |
| - | Subtraction |
| * | multiplication |
| / | Division |
| % | Modulus (remainder) |

// floor division

** exponent (power)

Example: a = 10
b = 3 ## output: 13.
print(a+b)

(ii) Assignment operators:

→ Assignment operators are used to assign or update values in a variable.
→ They combine arithmetic operation with assignment.
→ Used to assign values to variables

| Operator | Example | Output: 5 |
|---|---|---|
| = | x = 5 | |
| += | x += 3 | 8 |
| -= | x -= 2 | 6 |
| *= | x *= 2 | 12 |
| /= | x /= 2 | 6 |

(iii) Comparison operators

→ Comparison operators are used to compare two values
→ The result is always a boolean value (True (or) false)
→ Used to compare two values
→ Result is True or False

| Operator | Meaning |
|---|---|
| == | equal to |
| != | Not equal |
| > | Greater than |
| < | Less than |
| >= | Greater than or equal |
| <= | Less than or equal |

Example: a = 10
b = 20
print(a<b)

Output: True

(iv) Logical operators:

→ Logical operators are used to combine multiple conditions
→ They are commonly used in decision-making statements
→ Used to combine conditions.

| Operator | Meaning |
|---|---|
| and | True if both are True |
| or | True if any one is True |
| not | Reverses the result. |

Example:
```
a = 10
b = 20
print(a<b and b>15)
```
# Output: True

(v) Membership operators:

→ Membership operators check whether a value exists a sequence
→ Used with lists, strings, tuple and sets
→ Checks whether a value exists in a sequence

| Operator | Meaning |
|---|---|
| in | present |
| not in | not present |

Example:
```
list1 = [1, 2, 3]
print(2 in list1)
print('a' not in phython)
```

(vi) Identity operators:

→ Identity operators check whether two variables refer to the same memory object.
→ They do not compare values, but object identity.
→ check whether two variables refer to the same object.

| Operator | Meaning |
|---|---|
| is | same object |
| is not | Different object |

Example:
```
a = 10
b = 10
print(a is b)
```

## Real-world Usage of operators:

→ Calculate Salary
→ Check Eligibility
→ Validate login
→ Data analysis.

3. Explain phython Input and Output operations in detail

Input and output

→ Input is the data given to program by the user.
→ Output is the result produced by the program.
→ Input and output help is making programs interactive and dynamic.

Input in phython - input() function

Purpose of input()
→ Used to take input from the user.
→ Input is always read as a string.

Basic Syntax:

```
variable = input("message")
```

Example:

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

Output in phython - print() function.

Purpose of print()
→ Used to display output on the screen
→ can print text, numbers, variables and expressions

Basic Syntax:

```
print(value)
```

printing text:

```
print("Hello phython")
```

printing numbers and expressions

```
print(10)
print(5+3)
```

printing variables:

```
name = "prasad"
print(name)
```

printing multiple values:
→ use commas to seperate values

```
a = 10
b = 20   print("sum is", a+b)
```

New line in output:
→ Each print() creates a new line
→ \n is used for line break inside a string.

Type conversion while taking input:
Since input is always string, conversion is required.

Integer input:
```
age = int(input("Enter age:"))
print(age)
```

Float input:
```
price = float(input("enter price:"))
print(price)
```

Taking Multiple inputs:
Using seperate statements
```
a = int(input("enter a:"))
b = int(input("enter b:"))
```

Formatted output using print():
Using f-Strings (Recommended)
```
name = "prasad"
age = 21
print(f"Name: {name}, Age: {age}")
```

Separators:
```
print("phython", "Java", sep=",")
print("welcome", end=" ")
print("User")
```

#output: welcome User.

4. Discuss control statements and decision-making statements in Phython.

Control statements:

→ Control statements are used to control the flow of execution of a phython program.

→ They decide which statement is executed, how many times is executed or when execution stops.

→ Using control statements, we can write decision-making, looping and flow-altering programs.

Types of control statements in phython:

Phython control statements are broadly classified into three categories:

1. Decision making statements
2. Looping statements
3. Jump / control statements.

1. Decision making statements.

Decision making statements are used to execute code based on conditions. They depend on Boolean expressions (True/False)

(i) if statement

→ executes a block of code only if the condition is True

→ If the condition is false, the block is skipped.

Syntax: if condition:

    statement

Example: age = 20

    if age >= 18:

    print ("eligible to vote")

The message is printed only when age >= 18 is True.

(ii) if-else statement

→ executes one block if condition is True

→ executes another block if condition is false.

Syntax: if condition:
Statements
else:
Statements

Example: num = 5
if num % 2 == 0:
print("even number")
else:
print("odd number")

(ii) if - elif - else statement:
→ Used to check multiple conditions
→ elif means "else if"

Example: marks = 75
if marks >= 90:
print("Grade A")
elif marks >= 60:
print("Grade B")
else:
print("Grade C")

Use case: Grading systems, Menu-driven programs

(iv) Nested if statement
→ An if statement inside another if statement.
→ Used for complex decision-making.

Example: age = 20
citizen = "India"
if age >= 18:
if citizen = "India":
print("Eligible to vote").

5. Write an Essay on phython programming Fundamentals.

Phython is a popular programming language used for solving real-world problems efficiently. programming helps humans instruct computers to perform tasks such as calculations, data storage, automation, and application development.

Phython is known for its simple syntax and high readability, making it suitable for beginners. It uses English-like words and requires fewer lines of code compared to other languages.

Comments in Phython are used to explain code and improve documentation. They help programmers understand the logic and maintain programs easily.

Phython syntax is clear and close to the English language which reduces errors and improves productivity. Indentation is used instead of brackets, making the code neat and easy to understand. comments in phython (#) are used to explain code, making programs easy to maintain and debug.

Phython supports various data types such as integers, floats, strings and Booleans. operators perform calculations, comparisons, while input and output function allow users to interact with programs using input() and print().

Control flow statements like if, if-else, and loops helps the program make decisions and repeat tasks automatically. These fundamentals make phython a powerful and beginner-friendly language for developing applications.

## Real-world problem using Phython programming:

### 1. Movie Ticket pricing

A movie theatre 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) and prints the final ticket price.

Code:

```
age = int(input("Enter age: "))
is3D = int(input("Is it 3D movie? (1=yes, 0=no): "))
if age < 13:
    price = 150
elif age <= 59:
    price = 250
else: price = 200
if is3D == 1:
    price += 50
print("Final Ticket price: ₹", price)
```

Output:

```
Enter age: 12
Is it 3D movie? (1=yes, 0=no): 1
```

Final Ticket price: ₹200.

2. College Attendance Rule:

A student is allowed to write the exam if:

attendance $\geq 75$

OR

attendance $\geq 60$ AND has medical certificate (1 = yes, 0 = no)

Take attendance percentage and medical certificate as input and print "Allowed" or "Not allowed".

Code:

```
attendace = float(input("Enter attendance percentage: "))
medical = int(input("Medical certificate (1=yes, 0=no): "))
if attendance >= 75 or (attendance >= 60 and medical == 1):
    print("Allowed")
else:
    print("Not allowed")
```

Input:

Enter attendance percentage: 70

Medical certificate (1 = yes, 0 = no): 1

Output:

Allowed.

3. E-commerce Discount

A shopping site gives:
20% discount if bill ≥ 5000
10% discount if bill is between 2000 and 4999
No discount if bill < 2000
But if the customer is a prime member, they get extra 5% discount.
input: bill amount, is prime (1 or 0)
Print final amount to be paid.

Code:

```
bill = float(input("Enter bill amount: "))
is_prime = int(input("prime member? (1=yes, 0=no): "))
discount = 0
if bill >= 5000:
    discount = 20
elif bill >= 2000:
    discount = 10
if is_prime == 1:
    discount += 5
final_amount = bill - (bill * discount / 100)
print("Final amount to pay: ₹", final_amount)
```

Input:

```
Enter bill amount: 6000
prime member? (1=yes, 0=no): 1
```

Output:

```
Final amount to pay: ₹ 4500.0
```

4. Smart phone 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 message.
Input: Battery percentage, is charging (1 or 0)

Code:

```
battery = int(input("Enter battery percentage:"))
is charging = int(input("Is charging (1=yes, 0=no):"))
if is charging == 1:
    print("charging")
else:
    if battery <= 20:
        print("Low Battery")
    elif battery <= 80:
        print("Normal")
    else:
        print("Full")
```

Input:
Enter battery percentage: 85
Is charging (1=yes, 0=no): 0

Output:
Full

5. 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"

Code:

```
age = int(input("Enter age:"))
testpassed = int(input("passed driving test (1=yes, 0=no):"))
if age >=60:
    print("Eligible")
elif age >=18 and testpassed ==1:
    print("Eligible")
else:
    print("Not Eligible")
```

Input:

Enter age: 20

passed driving test (1=yes, 0=no): 1

Output:

Eligible

6. 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

Print "Free delivery" or "Delivery charged".

Code:

```
amount = float(input("Enter order amount: "))
is gold = int(input("Gold member (1=yes, 0=no): "))
distance = float(input("Enter distance : "))

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

Input:

Enter order amount: 450

Gold member (1=yes, 0=no): 1

Enter distance: 5

output:

Free delivery

7. Bank Loan Approval

A Bank approves a loan if:

Salary >= 30,000 AND Credit Score >= 700

OR

Salary >= 50,000 (Credit Score ignored)

Input: Salary, Credit Score

Print "Loan Approved" or "Loan Rejected"

Code:

```
Salary = float(input("Enter Salary: "))
CreditScore = int(input("Enter credit score: "))
if (Salary >= 30000 and CreditScore >= 700) or Salary >=
                                        50000:
    print("Loan Approved")
else:
    print("Loan Rejected")
```

Input:

Enter Salary: 55000

Enter CreditScore: 650

Output:

Loan Approved

8. 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

Code:

```
units = int(input("Enter units consumed: "))
bill = 0
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("Final bill amount: ₹", bill)
```

Input:

Enter units consumed: 250

Output:

Final bill amount: ₹ 850

9. Student Scholarship

A Student gets a scholarship 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)

Code:

```
marks = int(input("Enter marks: "))
income = int(input("Enter family income: "))
single parent = int(input("Single parent (1=yes, 0=no): "))
if marks >= 85 and (income < 500000 or single parent == 1):
    print("Scholarship Granted")
else:
    print("Scholarship not Granted")
```

Input:

Enter marks: 90

Enter family income: 600000

Single parent (1=yes, 0=no): 1.

Output:

Scholarship Granted

10. Online Exam Result

A student passes if:

theory ≥ 40 AND practical ≥ 40

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

Input: theory, practical

Code:

```
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")
```

Input:

Enter theory marks: 45

Enter practical marks: 50

Output:

pass

11. Hotel Room pricing

A Hotel charges:

₹ 3000 per day for normal days

₹ 4000 per day for 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("Is it weekend (1=yes, 0=no): "))
daysStayed = int(input("Enter number of days stayed: "))
if isWeekend == 1:
    price per day = 4000
else:
    price per day = 3000
total = price per day * daysStayed
if daysStayed > 3:
    total = total - (total * 15/100)
print("Final bill: ₹", total)
```

Input:

Is it weekend (1=yes, 0=no): 1

Enter number of days stayed: 4

Output:

Final bill: ₹ 13600.0

12. Graming level unlock

A game unlocks next level if:

Score >= 100

OR

Player has a premiumpass

But if player used cheating, accss is denied.

Input:- Score, isPremium, Usedcheat

Code:-

```
Score = int(input("Enter Score: "))
ispremium = int(input("premium pass (1=yes, 0=no): "))
Usedcheat = int(input("Used cheat (1=yes, 0=no): "))
 if Usedcheat == 1:
    print("Accss Denied")
 elif Score >= 100 or ispremium == 1:
    print("Next level unlocked")
 else:
    print("Level locked")
```

Input:

Enter Score: 120

Premium pass (1=yes, 0=no): 0

Usedcheat (1=yes, 0=no): 0

Output:

Next level unlocked.

13. Mobile data usage

A Network gives unlimited data if:

daily usage $\leq$ 2GB

OR

User has unlimited plan

But if roaming is on, unlimited plan does not work

Input: dataUsed, hasUnlimitedPlan, isRoaming

Code:

```
dataUsed = float(input("Enter data used today : "))
hasUnlimitedPlan = int(input("unlimited plan (1=yes, 0=no): "))
isRoaming = int(input("Roaming on (1=yes, 0=no): "))
if dataUsed <= 2:
    print("unlimited data")
elif hasUnlimitedPlan == 1 and isRoaming == 0:
    print("unlimited data")
else:
    print("Limited Data")
```

Input:

Enter data used today: 3

Unlimited Plan (1=yes, 0=no): 1

Roaming on (1=yes, 0=no): 0

Output:

Unlimited data

14. 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, facescan, isHoliday

Code:

```
idvalid = int(input("Is ID valid (1=yes, 0=no): "))
fingerprint = int(input("Fingerprint match (1=yes, 0=no): "))
facescan = int(input("Face scan match (1=yes, 0=no): "))
isHoliday = int(input("Is today Holiday (1=yes, 0=no): "))

if isHoliday == 1:
    print("Entry Denied")
elif idvalid == 1 and (fingerprint == 1 or facescan == 1):
    print("Entry Allowed")
else:
    print("Entry Denied")
```

Input:

Is ID valid (1=yes, 0=no): 1

Fingerprint match (1=yes, 0=no): 0

Face scan match (1=yes, 0=no): 1

Is today Holiday (1=yes, 0=no): 0

Output:

Entry Allowed.

15. Movie Rating Display

A movie app should rating based on 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 "Recommended".

Input: averageRating, isEditorsChoice (1 or 0)

Print the Message

Code:

```
averageRating = float(input("Enter average rating: "))
isEditorsChoice = int(input("Is Editors Choice (1=yes, 0=no): "))
if isEditorsChoice == 1:
    print("Recommended")
elif averageRating >= 8.5:
    print("Excellent")
elif averageRating >= 6.0
    print("Good")
else: print("Average")
```

Input:

Enter average rating: 7.8

Is Editors Choice (1=yes, 0=no): 0

Output:

Good.