

Assignment

1. Explain programming and python in detail.

Definition: Programming is the process of designing, writing, testing, and maintaining instructions that a computer follows to perform specific tasks.

Purpose of programming:

- To solve problems logically and efficiently
- To automate tasks and reduce manual effort
- To process data and generate useful information
- To develop software applications such as websites, mobile apps, games and operating systems.
- To control hardware like robots, sensors and embedded systems.

Characteristics of python:

- Simple and easy to learn
uses English-like syntax, making it beginner-friendly
- High-level language
programmers do not need to manage memory or hardware details.
- Interpreted language
code is executed line by line, making debugging easier.
- Object-oriented
supports classes and objects for modular and reusable code
- Portable
python programs can run on different platforms.
- Extensive library support
includes built-in and third party libraries for various tasks
- Open source
free to use and supported by a large community

Applications of python:

- Web development - Django, Flask
- Data science & analytics - Numpy, pandas, matplotlib
- Artificial Intelligence & machine learning - Tensorflow, scikit-learn
- Automation & scripting - Task automation
- Game development - pygame
- Scientific computing - research and simulations
- Desktop applications - Tkinter, PyQt
- IoT applications - Raspberry pi

Types of comments in python with syntax:

Comments are non-executable statements used to explain code and improve readability.

- Single-line comments: used to comment one line

Syntax: `# this is a single-line comment`
`print("Hello, Python")`

- Multi-line comments: python does not have a dedicated multi-line comment symbol, but we use

- a) Multiple single-line comments:

```
# this is line one  
# this is line two  
# this is line three
```

- b) Triple-quoted strings (used as comments):

"""

This is a multi-line comment
used for documentation

"""

Importance of python in modern software development:

- High demand language in IT Industry.
- used by top companies like Google, Amazon, Netflix
- Ideal for rapid application development
- Backbone of AI, ML and data science
- easy integration with other languages (C, C++, Java)
- Strong community support
- suitable for beginners and professionals
- widely used in education and research.

2. Describe data types and operators in python with suitable examples.

- Built-in data types in python

python provides several built-in data types to store different kinds of data.

a) Numeric data types: used to store numbers

- int - whole numbers

- float - decimal numbers

- complex - numbers with real and imaginary parts

Ex: $a = 10$ # int

$b = 3.5$ # float

$c = 2+4j$ # complex

b) Sequence data types: used to store multiple values in ordered manner

- string (str) - sequence of characters

- list - ordered and mutable collection

- tuple - ordered and immutable collection

Ex: $name = "Python"$ # string

$marks = [85, 90, 78]$ # list

$coordinates = (10, 20)$ # tuple

c) set data types: used to store unique and unordered elements

Ex: $unique_numbers = \{1, 2, 3, 4\}$

$\text{print}(unique_numbers)$ # output: {1, 2, 3, 4}

d) Mapping data types: used to store data in key-values pairs

- dictionary (dict)

Ex: $student = \{$

 "name": "Farheen",

 "age": 20,

 "marks": 88

e) Boolean data types: used to store true or false values

Ex: $is_pass = \text{true}$

$is_fail = \text{false}$

• Type identification using type():

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

Ex:

$x = 100$

$y = "Hello"$

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

`print(type(y))` # <class 'str'>

• Python operators:

Operators are the symbols used to perform operations on variables and values.

a) Arithmetic operators: used to calculate mathematical calculations.

operator	description
+	addition
-	subtraction
*	multiplication
/	division
%	modulus
**	exponent
//	floor division

Ex: $a = 10$

$b = 3$

`print(a+b)` # 13

`print(a%b)` # 1

b) Assignment operators: used to assign the values to the variables.

operator	example
=	$x = 5$
+=	$x += 3$
-=	$x -= 2$
*=	$x *= 4$
/=	$x /= 2$

Ex:
 $x = 10$
 $x += 5$
 $\text{print}(x)$ # 15

c) comparison operators: used to compare two values

operator	meaning
==	equal
!=	not equal
>	greater than
<	less than
>=	greater than or equal
<=	less than or equal

Ex:
 $a = 10$
 $b = 20$
 $\text{print}(a > b)$ # True

d) logical operators: used to combine conditional statements

operator	meaning
and	True if both conditions are true
or	True if at least one is true
not	Reverses the condition

Ex:
 $\text{age} = 18$
 $\text{print}(\text{age} > 18 \text{ and } \text{age} < 60)$
True.

e) membership operators: used to test whether a value is present in a sequence

operator	meaning
in	value exists
not in	value does not exist

Ex: fruits = ["apple", "mango"]
 $\text{print}("apple" \text{ in fruits})$ # True
 $\text{print}("grape" \text{ not in fruits})$
True

f) Identity operators: used to compare memory locations of objects

operator	meaning
is	same object
is not	different object

Ex: $a = 10, b = 10$

`print(a is b)` # True

`print(a is not b)` # False

- Real world usage of operators:

Arithmetic operators: Calculating salary, marks, discounts

Assignment operators: updating counters, scores, balances

Comparison operators: checking eligibility, pass/fail status

Logical operators: Authentication, decision-making

Membership operators: Searching data in lists or strings

Identity operators: Object comparison in memory-based programs.

3. Explain python input and output operations in detail.

Input and output (I/O) operations allows a program to receive data from the user and display results. Python provides simple and powerful functions for this purpose.

- `input()` function and its default data type:

It is used to take input from the user through the keyboard.

Syntax: `variable = input ("Message to user")`

Ex: `name = input ("Enter your name:")
age = input ("Enter your age:")
print(name)
print(age)`

- Type conversion while taking input:

Since `input()` returns a string, type conversion is required when performing numerical operations.

`int()` - converts to integer

`float()` - converts to decimal number

`str()` - converts to string

Ex: `age = int(input ("Enter your age:"))
salary = float(input ("Enter your salary:"))
print (age + 5)
print (salary * 2)`

- Taking multiple inputs:

Python allows taking multiple inputs in a single line using the `split()` method.

Ex: Multiple values in one line

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

- output using print() function:

It is used to display output on the screen.

Syntax: print(value1, value2,)

- a) using sep parameter: The sep parameter specifies how values are separated in output.

Ex: print ("Python", "Java", "C++", sep = ",") # Python, Java, C++

- b) using end parameter: The end parameter specifies what is printed at the end.

Ex: print ("Hello", end = " ")

print ("world") # Hello world

- c) formatted output using format(): The format() method is used to insert values into strings.

Ex: name = "Ahad"

marks = 85

print ("Name: {} , Marks: {}".format(name, marks))

- d) format specifiers (f-strings): Format specifiers control the appearance of output.

Ex: price = 49.56, 78

print(f"price: {price:.2f}") # 49.57

4. Discuss control statements & decision-making statements.

- Meaning of control statements:

Control statements are statements that control the flow of a program. Normally, a program runs line by line, but control statements allow us to change this flow based on conditions or repetition.

- Importance of control statements:

 - Help in decision making

 - Allow repetition of tasks

 - Reduce code duplication

 - Make programs logical and efficient

 - Help solve real-life problems like checking eligibility, grades.

- Types of control statements in python:

 - 1. Decision-making statements

 - 2. looping statements

 - 3. Jump statements

- Decision making statements:

They allow the program to choose one path based on a condition.

a) if statement: It executes a block of code only when the condition is true.

Syntax: if condition:
 statement

Ex: age = 18
 if age >= 18:
 print ("Eligible to vote")

b) if-else statement: It executes if one block if condition is true another block if the condition is false.

Syntax: if condition:
 statement
 else:
 statement

Ex: marks = 40
 if marks >= 35:
 print ("pass")
 else:
 print ("fail")

c) if-elif-else: It is used to check multiple conditions

Syntax: if condition:

```
    statement  
    elif condition:  
        statement  
    else:  
        statement
```

Ex: marks = 85

```
if marks >= 90:  
    print ("Grade A")  
elif marks >= 75:  
    print ("Grade B")  
elif marks >= 35:  
    print ("Grade C")  
else:  
    print ("Fail")
```

- Execution flow of decision making statements:

The program checks the condition.

If the condition is true, the corresponding block runs.

If the condition is false, it moves to the next block.

Only one block is executed.

Ex: checking exam result

Age eligibility for voting or driving

login validation

Electricity bill calculation

5. Write an essay on python programming fundamentals.

Programming plays an important role in solving problems in today's digital world. It helps humans communicate with computers by giving them clear instructions. Python is one of the most popular programming languages because it is simple, easy to learn and powerful. Python programming fundamentals form the base for developing useful and efficient programs.

Role of programming in problem solving:

Programming is the process of breaking a big problem into smaller steps and giving instructions to the computer to solve it. It helps in logical thinking and improves problem solving skills. Using programming, we can automate tasks, perform calculations, store data, and make decisions. Programming is used in many areas such as education, business, healthcare, banking, and technology.

Python syntax simplicity and readability:

Python is known for its simple syntax and easy readability. It uses English-like words and does not require complex symbols. Python programs are shorter and easier to understand compared to many other languages. Indentation is used instead of brackets, which makes the code clean and well-structured. Because of this simplicity, Python is ideal for beginners as well as professionals.

use of comments for code documentation:

Comments are used to explain the code and make it easier to understand. They are not executed by computer. Comments help programmers remember the logic of the program and allow others to understand the code easily. In Python, single-line comments are written using the # symbol, and multi-line comments can be

written using triple quotes. Good use of comments improves code readability, and maintenance.

Data types, operations and input/output operations:

Python provides various data types such as integers, floats, strings, lists, tuples, sets, dictionaries and boolean values to store different kinds of data. Operations are used to perform operations on data, such as arithmetic operations, comparisons and 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 using decision making statements:

Control flow statements decide the order in which statements are executed. Decision-making statements like if, if-else and if-elif-else allow the program to make decisions based on conditions. These statements help the program choose the correct path during execution. They are widely used in real-life applications such as checking eligibility, grading systems and validation processes.

Solve below real-world problems using python programming.

1. Movie ticket pricing.

```
age = int(input("Enter age:"))
is_3D = int(input("Is it a 3D movie? (1=Yes, 0=No):"))

# Base ticket price based on age
if age < 13:
    price = 150
elif age <= 59:
    price = 250
else:
    price = 200

# Add extra charge for 3D movie
if is_3D == 1:
    price += 50

print("Final ticket price: ₹", price)
```

Sample input/output:

```
Enter age: 65
Is it a 3D movie? (1=Yes, 0=No): 1
Final ticket price: ₹ 250
```

2. College attendance rule

```
attendance = 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")
```

Sample input/output:

Enter attendance percentage: 90
Medical certificate (1= Yes, 0= No): 1
Not allowed.

3. E-commerce discount

```
bill = float(input("Enter bill amount:"))  
isPrime = int(input("Is prime number? (1 for yes, 0 for no):"))  
discount = 0  
if bill >= 5000:  
    discount = 20  
elif bill >= 2000:  
    discount = 10  
if isPrime == 1:  
    discount += 5  
final_amount = bill - (bill * discount / 100)  
print("Final amount to be paid:", final_amount)
```

Sample input/output:

Enter bill amount: 6000
Is prime number? (1 for yes, 0 for no): 1
Final amount to be paid: 4500.0

4. Smartphone battery warning.

```
battery = int(input("Enter battery percentage:"))
ischarging = int(input("Is the phone charging? (1 for yes, 0 for no):"))

if ischarging == 1:
    print("charging")
else:
    if battery <= 20:
        print("Low Battery")
    elif battery >= 80:
        print("Normal")
    else:
        print("Full")
```

Sample input/output:

```
Enter battery percentage: 50
Is the phone charging? (1 for yes, 0 for no): 1
charging
```

5. Driving license check

```
age = int(input("Enter age:"))
testpassed = int(input("Passed driving test? (1 for yes, 0 for no):"))

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

Sample input/output:

Enter age : 19

Passed driving test ? (1 for yes, 0 for no) : 1
Eligible

6. Online food delivery

```
amount = float(input("Enter order amount :"))
```

```
isGold = int(input("Is gold member ? (1 for yes, 0 for no) :"))
```

```
distance = float(input("Enter delivery distance (in km) :"))
```

if distance > 10:

```
    print("Delivery not free")
```

```
elif amount >= 500 or isGold == 1:
```

```
    print("Free delivery")
```

else:

```
    print("Delivery not free")
```

Sample input/output:

Enter order amount : 600

Is gold member ? (1 for yes, 0 for no) : 1

Enter delivery distance (in km) : 10

Free delivery.

7. Bank loan approval

```
salary = int(input("Enter salary:"))
creditScore = int(input("Enter credit score:"))
if salary >= 50000 or (salary >= 30000 and creditScore >= 700):
    print("Loan approved")
else:
    print("Loan rejected")
```

Sample input/output:

```
Enter salary: 85000
Enter credit score: 750
Loan approved
```

8. Electricity bill

```
units = int(input("Enter units consumed:"))
```

```
bill = 0
```

```
if units <= 100:
```

```
    bill = units * 2
```

```
elif units <= 300:
```

```
    bill = (100 * 2) + (units - 100) * 3
```

```
else:
```

```
    bill = (100 * 2) + (100 * 3) + (units - 200) * 5
```

```
-print("Final bill amount: ₹", bill)
```

Sample input/output:

```
Enter units consumed: 80
Final bill amount: ₹160
```

9. Student Scholarship

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

Sample input/output:

```
Enter marks : 50
Enter family income : 400000
Single parent child? (1 for yes, 0 for no) : 1
Scholarship granted
```

10. Online exam result

```
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")
```

Sample input/output:

```
Enter theory marks : 75
Enter practical marks : 58
Pass
```

11. Hotel room pricing

```
isweekend = int(input("Is it a weekend? (1 for yes, 0 for no):"))
daysStayed = int(input("Enter number of days stayed:"))
if isweekend == 1:
    rate = 4000
else:
    rate = 2000
total = rate * daysStayed
if daysStayed > 3:
    total = total - (total * 0.15)
print("Final bill amount: £", total)
```

Sample input/output:

```
Is it a weekend? (1 for yes, 0 for no): 0
Enter number of days stayed: 20
Final bill amount: £ 7600.0
```

12. Gaming level unlock

```
score = int(input("Enter score:"))
isPremium = int(input("Has premium pass? (1 for yes, 0 for no):"))
usedcheat = int(input("Used cheating? (1 for yes, 0 for no):"))
if usedcheat == 1:
    print("Access denied")
elif score >= 100 or isPremium == 1:
    print("Next level unlocked")
else:
    print("Level locked")
```

Sample input/output:

Enter score: 89

Has premium pass? (1 for yes, 0 for no): 1

used cheating? (1 for yes, 0 for no): 0

next level unlocked

13. Mobile data usage

```
dataUsed = float(input("Enter data used today (in GB):"))
hasUnlimitedPlan = int(input("Has unlimited plan? (1 for yes, 0 for no):"))
isRoaming = int(input("Is roaming on? (1 for yes, 0 for no):"))

if isRoaming == 1:
    if dataUsed <= 2:
        print("unlimited data")
    else:
        print("limited data")
else:
    if dataUsed <= 2 or hasUnlimitedPlan == 1:
        print("unlimited data")
    else:
        print("limited data")
```

Sample Input/output:

Enter data used today (in GB): 3

Has unlimited plan? (1 for yes, 0 for no): 0

Is roaming on? (1 for yes, 0 for no): 1

Limited data.

14. Office entry system

```
idValid = int(input("Is ID card valid? (1 for yes, 0 for no):"))
fingerprint = int(input("Fingerprint matches? (1 for yes, 0 for no):"))
faceScan = int(input("Face scan matches? (1 for yes, 0 for no):"))
isHoliday = int(input("Is it a holiday? (1 for yes, 0 for no):"))

if isHoliday == 1:
    print("Entry denied")
elif idValid == 1 and fingerprint == 1 or faceScan == 1:
    print("Entry allowed")
else:
    print("Entry denied")
```

Sample input/output:

```
Is ID card valid? (1 for yes, 0 for no): 1
Fingerprint matches? (1 for yes, 0 for no): 1
Face scan matches? (1 for yes, 0 for no): 1
Is it a holiday? (1 for yes, 0 for no): 0
Entry allowed
```

15. Movie rating display,

```
averageRating = float(input("Enter average rating:"))
isEditorChoice = int(input("Is editor's choice? (1 for yes, 0 for no):"))

if isEditorChoice == 1:
    print("Recommended")
else:
    if averageRating >= 8.5:
        print("Excellent")
    elif averageRating >= 6.0:
        print("Good")
    else:
        print("Average")
```

Sample input/output:

```
Enter average rating: 9.3
Is editor's choice? (1 for yes, 0 for no): 0
Excellent
```