

Python Assignment

1. Explain Programming and Python in detail

Definition :-

Programming is the process of writing instructions that a computer follows to perform a task.

Purpose of Programming :-

* Problem solving :- To find solutions to real-world problems using logic and algorithms.

* Computer cannot think on their own.

* They only understand instructions.

* Programs help automate tasks and solve problems.

Real World Examples :-

Calculator → performs calculations.

ATM → processes transactions.

Mobile phones → respond to user actions.

Programming Languages :-

C, C++, Java, Python, JavaScript

Definition of Python :-

Python is a high level, interpreted, general purpose programming language.

Characteristics of Python :-

* Simple and Easy to learn :- Python has Simple Syntax, which is easy to read and write.

* Interpreted Language :- Python is an Interpreted language because in python code is executed line by line, making debugging easy.

* Platform Independent :- Code can run on any Operating System without any modification.

* Open Source :- Python is free and open Source that is code can easily visible, available.

* Large Library :- Python contains many built-in libraries.

* Object Oriented :- Python is an Object Oriented language it supports classes and objects.

Applications of Python :-

* Data Science and Analytics

* Web Development

* Artificial Intelligence

* Machine Learning

* Automation

* Game development

* Desktop Applications

Comments in Python :-

Comments are non-executable lines used for explanation.

* Comments are ignored by Python.

* Used to explain code

* Makes program readable and understandable

Python supports two types of Comments

Types of Comments

* Single Line Comment

* Multi Line Comment

* Single Line Comment

Syntax:-

This is a Single Line Comment.

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

* Multi Line Comment

Syntax:-

'''This is

Multiline

Comment'''

Importance of Python in Modern Software Development

Development :-

1) Easy to learn and use :-

Python has a simple and readable syntax, which makes it easy for beginners to learn.

2) Wide Range of Applications :-

* Python is used in many domains such as Web Development, Data Science and analytics; AI, ML etc.

3) Strong Libraries and Frameworks :-

* Python has rich collection of libraries and frameworks like Numpy, Pandas, Tensorflow etc.

4) High Demand in Industry :-

Many top companies like Google, Netflix, Spotify etc use Python.

2. Describe Data Types and Operators in Python

with suitable examples.

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

Classification of Data Types :-

Python data types are mainly classified into

- 1.) Numeric
- 2.) Text
- 3.) Sequence
- 4.) Set
- 5.) Mapping
- 6.) Boolean

1) Numeric data type :-

It includes int, float, complex

Int :- Stores whole numbers

Float :- Stores decimal numbers

Positive / Negative

Ex :- $x = 15.8$

$= a = 10$

$y = 6.5$

$b = -8$

Complex

* Stores numbers in $a + bj$ format

$z = 2 + 5j$

2) Text Data Type :-

str :- It stores text or characters

* It will be written inside quotes

Ex :- name = "xyz"

colour = "Green"

* It can be written by single quotes or double quotes

Sequence Data Types

list :-

- * Ordered collection
- * Mutable (can change)

Ex:-

numbers = [1, 2, 3, 4]

Range :-

- * Represents Sequence of numbers.
- * Commonly used in loops

Ex:-

r = range(1, 5)

4) Set Data Type

set :-

- * Unordered collection
- * No duplicate values
- * It stores only unique values.

Ex:- S = {5, 10, 15, 20, 25}

5) Mapping Data Type :-

dict :-

- * Stores values data in key-value pairs
- * Key values must be unique.

Ex:- student = {"name": "Abhi", "class": 5}

Accessing a value :

print(student["name"])

6) Boolean

- * Stores only True or False

- * Used in decision making statements

Ex:- is_valid = TRUE

Tuple :-

- * Ordered collection
- * Immutable (cannot change)

Ex:-

colors = ("red", "pink", "blue")

- * Type identification using `Type()`
- * The `Type()` function in Python tells what kind of data an object is.
- * It is mainly used to identify data types at run time.

Syntax

`type(Object)`

`x =`

`X = 10`

`print(type(x))`

Output :- `int`

Various Python Operators (Arithmetical Operators, Comparison Operators, Assignment, Logical, Bitwise, Membership, Identity)

* Arithmetic Operators :-

* Arithmetic Operators are used to perform mathematical calculations.

* They work with members like integers and floats

Operator	Meaning
<code>+</code>	Addition
<code>-</code>	Subtraction
<code>*</code>	Multiplication
<code>/</code>	Division
<code>%</code>	Modulus
<code>//</code>	Floor Division
<code>**</code>	Exponent

`x =`

`a = 20`

`b = 5`

`print(a**b)`

* Comparison Operators:-

- * It is used to compare two values.
- * It gives Results as True or False

Operators	Meaning
$= =$	Equal to
$<$	Less than
$>$	Greater than
\geq	Greater than or equal to
\leq	Less than or equal to
$!=$	Equal to

Ex :-

$$a = 20$$

$$b = 50$$

$$\text{print}(a < b)$$

* Assignment Operators :-

- * It is used to assign values to variables

Operators	Meaning
$=$	Add
$+ =$	Add & Assign
$- =$	Subtract & Assign
$* =$	Multiply & Assign
$/ =$	divide & Assign
$\ =$	floor & Assign
$\% =$	Modulus & Assign
$** =$	Power & Assign

Ex :-

$a = 15$

$a + b$

print(a)

* Logical Operators :-

* Logical Operators is used to combine conditional statements

Operators	Meaning
and	True if both are True
or	True if anyone is True
not	Reverse the result

Ex :- $a = 10$

$b = 30$

print($a < b$ and $b > a$)

* Membership :- Used to test membership in a sequence

Operator	Meaning
in	Present
not in	Not Present

Ex :- text = "python"

print('p' in text)

* Identity Operator :-

* It is used to compare memory locations

Operator	Meaning
is	Same object
is not	Different object

Ex :- $x = 30$

$y = 30$

print(x is y)

Real World Usage of Operators

Operators

Arithmetic

Assignment

Comparison

Logical

Membership

Identity

Bitwise

Real World Use

Billing, Salary, Marks

Updating values

Eligibility checks

Multi-condition rules

Search & validation

None / Null checks

Security and Networking

3. Explain Python Input and Output Operations in detail.
Include the following in your explanation.

* Input() function and its default data type

* Input() function is used to take input from the user through the key board

Syntax : Variable = input ("Enter text")

Ex : * name = input ("Enter your name")
print(name)

* age = input ("Enter your age")

print(type(age))

* Type Conversion while Taking Input

Since input() returns a string, we must convert it into required data types (int, float etc).

Type Conversions:

* int() = converts to integer

* float() = converts to float i.e. decimal value

* str() = converts to string

Ex: Int to:

age = int(input("Enter age:"))

print(age + 5)

Output will be 25 if user enter 20

Float Example:

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

print(price)

Output price+9.5

Enter price: 49.5

49.5

Taking Multiple Inputs:

* Python allows taking multiple inputs in one line using the split() method.

Syntax: a, b = input().split()

Ex:

name, age = input("Enter name and age").split()

print(name, age)

Formatted Output:

* Using print():

Ex: print("Hello", name)

Using Separators:

- * It is used to separate the values in output

Ex:

```
print("python", "Java", "C", sep = ",")
```

Output:

Python, Java, C

End Parameter:

It specifies what is printed at the end.

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

```
print("World")
```

Output:

Hello World

Format Specifiers:

- * It is used to insert values into a string in a structured way.

Ex:

```
name = "Anu"
```

```
age = 20
```

print(f "My name is {name} and age is {age}")

Statements

4. Discuss Control Statements and Decision-making

in Python.

Meaning of Control Statements:

Definition: Special instructions that manage the flow of execution in a program, determining the sequence in which other instructions are executed.

Importance of Control Statements:-

- 1) Decision making:-
 - * Allows programs to take decisions based on conditions
Ex:- checking age eligibility, discount calculation etc.
- 2) Reduces Code Repetition:
 - * Loops help run the same code multiple times without rewriting it.
- 3) Improves program efficiency:
 - * Executes only required code, saving time and resources.
- 4) Essential for Real-World applications:
 - * Used in login systems, billing system, banking apps, exams etc.

Types of Control Statements:-

- * Decision-making
- * Looping
- * Jumping | Control Transfer Statements

Decision-Making Statements in Python

- * Decision-Making Statements are used to execute code based on conditions.
- * They depend on Boolean Expressions

1) If Statement:-

- * if Statement is used to execute a block of code only when condition is true

Syntax:

if condition:
 statement

Ex:

age = 20

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

3) if-else Statement

Used to execute one block if the condition is true
and another block if it is false

Syntax:

if condition:
 Statement 1
else:
 Statement 2

Ex:

num = 6

if num % 2 == 0:
 print("Even number")
else:
 print("Odd number")

3) if-elif-else Statement

* It is used to when we need to check
multiple conditions

Syntax: if condition 1:
 Statement 1
elif condition 2:
 Statement 2
else:
 Statement 3

Ex:

```
marks = 72  
if marks >= 90:  
    print("Grade A")  
elif marks >= 60:  
    print("Grade B")  
else:  
    print("Grade C")
```

4) Nested if statement

- * A nested if statement is an if statement placed inside another if statement.
- * We can use nested if statements to check conditions within conditions.

Syntax:

```
if condition1:  
    if condition2:  
        statement  
    else:  
        statement  
else:  
    statement
```

Ex:

```
age = 22  
test = 1  
if age >= 18:  
    if test == 1  
        print("Eligible for license")  
    else:  
        print("Test not passed")  
else:  
    print("Underage")
```

Syntax flow and execution Control

- * Conditions are evaluated from top to bottom
- * As soon as a condition becomes True, its block is executed.
- * Remaining conditions are skipped.
- * If no condition is true, the else block is executed.

5. Write an essay on Python Programming fundamentals.

- * Role of programming in problem solving
- * Programming helps converts real-world problems into logical steps that a computer can execute.
- * It automates tasks, reduces errors, and improves efficiency.
- * Example includes calculating bills, checking eligibility, analyzing data.

* Python Syntax Simplicity and Readability

- * Python uses simple and English-like syntax, making it easy to learn and read.

- * Less punctuation and indentation based blocks improve clarity.

Ex :-

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

- * Use of Comments for Code Documentation :-
- * Comments explain code for better understanding and maintenance

In python Comments includes :-

- * Single line Comments :-

~~Ex:-~~

```
# This is a single-line comment  
print("Hello world")
```

- * Multi - Line Comments :

```
''' This is  
a multiline  
comment'''
```

- * Data Types, Operators and Input / Output Operations

Data Types :

- * It includes int, float, string, boolean
- * And list, tuple, dict also included in Python

Data Types .

- * List is mutable and it is ordered collection

- * Tuple is immutable .

* Dic stores values data in key-value pairs

Operators : +, -, *, /, >, <, ==, and, or

Input / output Example :-

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

```
print ("Hello", name)
```

- ★ Control Flow Using Decision-Making Statements
- * Conditional statements like if, if-else, if-elif-else, and nested if allow programs to make decisions based on input or conditions
- * Example: Determining eligibility, grade, discount,

Example :-

age = 20

if age >= 18 :

 print ("Eligible to vote")

else:

 print ("Not eligible to vote")

- * Real-World problems often have choices or conditions

- * Decision-making statements help programs respond differently based on input

- * Examples includes :

- * Calculating discounts based on bill amount.

- * Determining if someone is eligible to vote

Real World Problems using Python Programming

- 1.) Movie Ticket Pricing :- Write a program that takes age and is3D(1 or 0) and prints the final ticket price.

```
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:
```

```
    print += 50
```

```
print("Final price of a ticket:", price)
```

- 2.) Take Attendance percentage and medical certificate as input and print "Allowed" or "Not Allowed".

```
present = int(input("Enter Attendance percentage:"))
```

```
medical = int(input("Medical certificate (1=yes, 0=no):"))
```

```
if present >= 75 or (present >= 60 and medical == 1):
```

```
    print("Allowed")
```

```
else:
```

```
    print("Not Allowed")
```

3. E-Commerce Discount

A Shopping site gives :-

```
bill = float(input("Enter Bill amount:"))
```

```
isPrime = int(input("Prime number (1=Yes, 0=No)"))
```

```
discount = 0
```

```
if bill >= 5000:
```

```
    discount = 20
```

```
elif bill >= 2000:
```

```
    discount = 10
```

```
else:
```

```
    discount = 0
```

```
If isPrime == 1:
```

```
    discount = discount + 5
```

```
final_amount = bill - (bill * discount / 100)
```

```
print("Final amount to be paid:", final_amount)
```

4. Smartphone Battery Warning

A phone shows :-

```
battery = int(input("Enter battery percentage:"))
```

```
isCharging = int(input("Is charging (1=Yes, 0=No)"))
```

```
if isCharging == 1:
```

```
    print("Charging")
```

```
elif battery <= 20
```

```
    print("Low Battery")
```

```
elif battery >= 80
```

```
    print("Normal Battery")
```

```
else:
```

```
    print("Full")
```

5. Driving license Check:

A person can get a driving license if:

```
age = int(input("Enter age :"))
test = int(input("Driving test passed (1=yes, 0=no) :"))

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

6. Online-Food Delivery

A restaurant gives free delivery if

```
amt = int(input("Enter amount"))
isGold = int("User is Gold member (1=yes, 0=no)"))
dis = float(input("Enter distance in km:"))

if dis > 10:
    print("Delivery is charged")
elif amt >= 500 or isGold == 1:
    print("Free Delivery")
else:
    print("Delivery charged")
```

7. Bank Loan Approval

7) bank approves a loan if \div

Sal = int(input("Enter salary :"))

CS = int(input("Enter credit score"))

if (Sal \geq 30000 and CS \geq 700) or Sal \geq 50000:
 print("Loan Approved")

else:

 print("Loan Rejected")

8. Electricity Bill :

Units Consumed :

units = int(input("Enter units Consumed"))

if unit \leq 100 :

 bill = units * 2 :

elif unit \leq 200 :

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

else:

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

print("Final Bill amount : ", bill)

9. Student Scholarship :-
A Student gets a Scholarship if :-
 $m = \text{int}(\text{input}("Enter marks"))$
 $\text{income} = \text{int}(\text{input}("Enter income of student family"))$
 $\text{singleparent} = \text{int}(\text{input}("Is Single parent (1=yes, 0=no)"))$
if $m \geq 85$ and ($\text{income} < 50000$ or $\text{singleparent} == 1$):
 print ("Scholarship Granted")
else:
 print ("Scholarship Not Granted")

10. Online Exam Result :

$\text{theory} = \text{int}(\text{input}("Enter theory marks :"))$
 $\text{practical} = \text{int}(\text{input}("Enter practical marks :"))$
if ($\text{theory} \geq 40$ and $\text{practical} \geq 40$) or
 ($\text{theory} + \text{practical} \geq 100$):
 print ("pass")
else:
 print ("Fail")

11. Hotel Room Pricing :-

A hotel charges:

```
week = int(input("Is it weekend (1=yes, 0=no)"))  
days = int(input("Days stayed:"))  
if isWeek == 1:  
    cost = 2000 * days  
else:  
    cost = 3000 * days  
if days > 3:  
    cost = cost - (cost * 15 / 100)  
print("Final Bill:", cost)
```

12. Gaming Level Unlock:

A game unlocks next level if :

```
scr = int(input("Enter score:"))  
isPremium = int(input("Premium pass (1=yes, 0=no)"))  
usedCheat = int(input("Used cheating (1=yes, 0=no)"))  
if usedcheat == 1:  
    print("Access Denied")  
elif scr >= 100 or isPremium == 1:  
    print("Next Level Unlocked")  
else:  
    print("Level Locked")
```