

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:

- Computers cannot think on their own.
- They only understand instructions
- Programs help automate tasks and solve problems.
- Purpose of programming is to make human life better and easier.
- It does so because the computer programs can perform a set of tasks once we define them clearly.

Python;

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

Characteristics of Python:

- simple and easy to learn
- interpreted language
- high-level language
- Dynamically Typed
- object-oriented
- portable
- open source and free

applications of python:

- > Data science & Analytics
- > web development
- > Artificial intelligence
- > machine learning
- > Automations
- > 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 have two types of comment lines:

- > single line comment
- > multi line comment.

Single line comment:

- > written using the # symbol.
- > it uses double quotes.

e.g.: print "Hi, Hello"

Multi line comment:

- > it also written in # symbol.
- > it uses triple quotes.

e.g.: print """welcome to party""".

importance of python in modern software development.

Python has become one of the most important and widely used programming languages in modern software development.

- Easy to learn and use
- High productivity
- Versatile and multi-purpose
- Rich libraries and frameworks
- Strong community support
- Platform independent.

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

Data types specifies what kind of data a variable holds, python determines data types automatically.

Numeric data type:

Int

- Stores whole numbers
- positive or negative

$a = 10$

$b = -5$

float

- Stores decimal numbers

$\pi = 3.14$

$y = 10.5$

Complex:

→ stores numbers in a+bj format

$$z = 2 + 3j$$

Sequence Data type:

list

- ordered collection
- mutable (can change)

marks = [80, 85, 90]

tuple

- ordered collection
- immutable (cannot change)

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

range

- represents sequence of numbers

r = range(1, 5)

Set Data type:

- unordered collection
- No duplicate values.

s = {1, 2, 3}

mapping Data type:

dict

- stores data in key-value pairs

e.g. - details = {"name": "madhavi", "age": 20, "Branch": "stats"}

print(details).

Boolean Data type

bool:-

- Stores only True or False
 - used in decision making
- is_valid = True.

Type identification using type()

$a = 10$

Print(type(a))

Python operators:

Arithmetic operator:-

- Arithmetic operator are used to perform mathematical calculations.
- use for mathematical calculations.

operator	meaning
+	Addition
-	Subtraction
*	multiplication
/	Division
%	modulus (remainder)
//	Floor division
* *	Exponent (power)

Eg:

$a = 5$

$b = 2$

Print(a+b), print(a-b), print(a*b)

comparison operator

- comparison operators are used to compare two values.
- The result is always a boolean value (true or false)
- used to compare two values.

operator	meaning
----------	---------

$= =$	Equal to
\neq	not equal
$>$	Greater than
$<$	Less than
\geq	Greater than equals to
\leq	Less than equals to.

Ex:- $a=5$ $b=3$

Point ($a > b$)

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.

Ex:-

$$a = 10$$

$$b = 20$$

Print (a < b and b > 15)

Assignment operator..

- The 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 .

$$= \quad x = 5$$

$$+ = \quad x + = 3$$

$$- = \quad x - = 2$$

$$* = \quad x^* = 2$$

$$/ = \quad x / = 2$$

Ex:-

$$x = 10$$

$$x + = 5$$

Print (x)

membership operator:-

- membership operator check whether a value exists in a sequence
- used with lists, strings, tuples and sets
- check whether a value exists in a sequence.

operator meaning

in present

not in not present

Ex:-

list1 = [1, 2, 3]

Print (2 in list1)

Print ('a' not in 'python')

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 same object.

operator meaning

is same object

is not different object

Ex:- a=10

b=10

Real world operators:

Arithmetic operators: used in billing systems, banking apps, games.

Comparison operators: used in decision-making, validations, eligibility checks.

Logical operators: - used in authentication systems, form validation, access control.

Assignment operators: - used in updating values, counters, totals.

Membership operators: - used in searching, filtering data.

Identity operators: - used in object comparison, memory management.

3) Explain python input and output operations in details.

- Input is the data given to a program by the user.
- Output is the result produced by the program
- Input and output helps in making programs interactive and dynamic.

input() function..

- used to take input from the user.
- input is always read as a string.

Syntax:-

variable = input ("message")

Integer datatype:-

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

Float input datatype:-

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

Type conversion while taking input

int() Integer

float() float

str() string

bool() boolean.

Taking multiple inputs

Using separate statements

```
a = int (input ("Enter a:"))
b = int (input ("Enter b:"))
print (a)      print (b)
```

format specifiers:

Using f-strings

name = "madhavi"

age = 20

```
print (f "Name: {name}, Age : {age}")
```

4) Discuss control statements and decision-making statements in python.

- control statements are used to control the flow of execution of a python program.
- They decide which statement is executed how many times it is executed, how many times it is executed, or when execution stops.
- using control statements, we can write decision-making, looping and flow-altering programs.

Types of control statements:-

- Decision making statements
- Looping
- Jump/ control transfer statements.

Decision making statements:

Decision making statements are used to execute code based on conditions. They depend on boolean expressions (true | false)

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.

Ex:- age = 20

if age >= 18;

Print ("Eligible to vote")

if-else statement:-

- Executes one block if condition is true.
- Executes another block if condition is false.

Syntax:-

if condition:

Statements

else:

Statements.

Ex:-

num = 5

if num % 2 == 0,

Print ("Even number")

else

Print ("Odd number")

if-elif-else Statement

- Once a condition is true, remaining conditions are skipped.

Ex:- marks = 85

- elif means else if.

Syntax:-

if condition 1: Statement

Statement else:

elif condition 2: Statement

Statement

elif condition 3:

if marks >= 90:

Print ("Grade A")

elif marks >= 95:

Print ("Grade B")

else:

Print ("Grade C")

5) write an essay on python programming fundamentals:-

Python programming fundamentals include variables, data types, control statements, functions etc.

Key fundamentals:

Variables and Data types:-

Variables are used to store information in python.

Python is dynamically typed so we don't need to declare data types. common data types are.

- Integer
- float
- Strings
- Boolean

Operators:-

Operators are symbols which are used to perform operations on values and variables. These include arithmetic, assignment and comparison operators

Arithmetic operations (+, -, *, /, //)

Assignment operations (=, +=, -=)

Comparison operators (==, !=, >, <)

Control statements:-

Statements that control the flow of execution in program.

Decision making statements like if, else if else

elif

Looping statements like for and while loops.

functions:-

Reusable blocks of code that perform a specific task is known as function. They are defined using the keyword "def" and accept parameters and returns values.

Input and output:-

functions like print() display output to the screen, whereas input() function gets data from the user via the keyboard.

Programs

1) movie ticket pricing:

```

age = int(input("Enter age:"))
is3D = int(input("3D movie? (1=Yes, 0=No):"))

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

if is3D == 1:
    Price = price + 50

print("Ticket price =", Price)

```

Output

Enter age: 45
3D movie? (1=Yes, 0=No): 1
Ticket price : 300

2)

College Attendance Rule

```

a = int(input("Enter the a:"))
m = int(input("Enter the m:"))

if a >= 75 or (a >= 60 and m == 1):
    print("Allowed")
else:
    print("Not allowed")

```

Output

Enter the a : 150
Enter the m : 160
Allowed

3) E-commerce Discount

```

b = int(input("Enter the b:"))
c = int(input("Enter the c:"))

if b > 5000:
    b = b - b * 0.20
elif b >= 2000:

```

Output

Enter the b : 3500
Enter the c : 5600
3150.0

$b = b - b * 0.10$

if

$c = 1;$

$b = b - b * 0.05$

Print (b)

4) Smart phone Battery warning:

$m = \text{int}(\text{input}("Enter the m:"))$

$K = \text{int}(\text{input}("Enter the K:"))$

if $K:$

Print ("charging")

elif

$m <= 20;$

output

Print ("Low battery")

Enter the m : 21

elif

$m <= 80:$

Enter the K : 68

Print ("Normal")

charging

else :

Print ("Full")

Driving license check

$age = \text{int}(\text{input}("Enter the age:"))$

$testpassed = \text{int}(\text{input}("Enter the testpassed:"))$

if $age >= 60$ or ($age >= 18$ and $testpassed = 1$):

Print ("Eligible")

output

else :

Print ("not Eligible")

Enter the age : 76

Enter the testpassed : 85

Eligible.

6) Online Food Delivery

```
S = int(input("Enter the S:"))
K = int(input("Enter the K:"))
g = int(input("Enter the g:"))

if g <= 10 and (g >= 500 or K == 1):
    print("Free")
else:
    print("paid")
```

Output

Enter the S : 56
Enter the g : 98
Enter the K : 97

"paid"

7) Bank loan Approval

```
K = int(input("Enter the K:"))
S = int(input("Enter the S:"))

print("loan approved") if (K == 30000 and S >= 700)
    or K >= 50000 else "loan Rejected"
```

Output

Enter the K : 200000

Enter the S : 600000

"loan Approved"

8) Electricity bill

```
U = int(input("Enter the U:"))
```

if U <= 100:

$$\text{bill} = U * 2$$

elif U <= 200:

$$\text{bill} = 100 * 2 + (U - 100) * 3$$

else:

$$\text{bill} = 100 * 2 + 100 * 3 + (U - 200) * 5$$

Print(bill)

Output

Enter the U : 98

196

9) Student scholarship ship

```
marks = int(input("Enter the marks:"))
income = int(input("Enter the income:"))
singleparent = int(input("Enter the singleparent:"))

if marks >= 85 and (income < 50000 or singleparent == 1):
    print("Scholarship Granted")
else:
    print("not Eligible")
```

Output

Enter the marks : 95
Enter the income : 60000
Enter the singleparent : 1
"scholarship Granted"

10) Online Exam result

```
theory = int(input("Enter the theory:"))
Practical = int(input("Enter the practical:"))

if (theory >= 40 and Practical >= 40) or (theory + Practical >= 100):
    print("Pass")
else:
    print("Fail")
```

Output

Enter the theory : 60
Enter the practical : 50
"Pass"

1) Hotel room pricing.

w = int(input("Enter the w:"))

Output

d = int(input("Enter the d:"))

Enter the w: 2000

b = int(in

Enter the d : 3000

b = (4000 if w else 3000) * d

b - 102000000

Print(b * 0.85 if d > 3 else b)

2) Gaming level unlock

score = int(input("Enter the score:"))

ispremium = input() == "yes"

usedcheat = input() == "yes"

if(score >= 100 or ispremium) and not usedcheat:

 Print("level unlocked")

else:

 Print("Access Denied")

Output

Enter the score : 120

NO

NO

level unlocked

mobile data usage

dataused = 1.5

hasunlimitedPlan = True

isRoaming = False

if(dataused <= 2 or hasunlimitedplan) and not isRoaming:

 Print("unlimited data")

else:

 Print("No unlimited Data")

Output

unlimited Data

14) Office Entry System

```
I = int(input("Enter 1 for ID valid else 0;"))
F = int(input("Enter 1 for fingerprint match else 0;"))
fscan = int(input("Enter 1 for facescan match else 0;"))
isholiday = int(input("Enter 1 for holiday else 0;"))

if isholiday == 1:
    print("Entry denied")
elif f == 1 and (f == 1 or fscan == 1):
    print("Entry allowed")
else:
    print("Entry denied")
```

Output:-

```
I : 1
F : 0
fscan : 1
Holiday : 0
Entry allowed.
```

15) Movie Rating Display

```
rating = float(input("Enter the rating;"))
editor = int(input("Enter the editor;"))

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

Output

```
Enter the rating : 9.5
Enter the editor : 1
Recommended.
```