

Assignment

(1) Explain the programming and python in detail.

A:-

Definition :-

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

→ Purpose :-

- i) They only understand instructions.
- ii) Programs help automate tasks & solve problems
- iii) Computers cannot think on their own.

→ Comments used to explain code and non-executable lines used for explanation.

→ Types in comments

(1) Single line comment

Syntax: - # This is single line comment

(2) Multi-line comment

Syntax:-

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

Importance of python:-

- Platform Independent
- Large library support
- Simple & easy to learn
- High level interpreted language.

(Q) Describe Data Types and Operators in python with suitable examples.

Data types specifies what kind of data variable holds and determines data type automatically.

Data types are classified into:-

(1) Numeric data type

→ int

- store whole numbers
- positive or negative

a = 10

b = -8

→ float

- stores decimal numbers

m = 7.14

n = 0.9

(2) Complex

- stores numbers in $a+bj$ format

$z = 2+3j$

(3) Text Data type

→ str

- stores text or characters
- written inside quotes

name = "zebra"

num = "one"

(4) Boolean Data Type

→ bool

- stores only true or false
- used in decision making

is_valid = True

(5) Sequence Datatypes

→ list

- ordered collection
- mutable (changes)

grades = [75, 96, 84]

(5) Set data type

→ set

- unordered collection
- no duplicate values

S = {1, 2, 3}

→ tuple

- ordered collection
- immutable (cannot change)

shades = (maroon, beige, gold)

(6) Mapping data types

→ dict

- Stores data in key-value pairs

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

→ Identification checked using type() function

x = 9

print(type(x))

→ Various Python Operators are categorized in following:-

(i) Arithmetic operators:-

- These are used to perform mathematical calculations and numbers like integers.

Operators

+

→ addition

/ → division

-

→ subtraction

% → modulus (remainder)

*

→ multiplication

// → floor division

** → Exponent (power)

(ii) Comparison operators:-

- These operators used to compare & values and result always a boolean value (T or F)

Operators

==

→ Equal to

>= Greater than or equal

!=

→ Not equal

<= Less than or equal

>

→ Greater than

<

→ Less than

- Usage of operators in real world
 - Total bill, tax
 - Average score
 - Age verification
 - Filtering data

(3) Explain Python Input and output operations in detail.

→ Input is the data given to a program by the user.
 Input and output help in making programs
 interactive and dynamic

• Input() function

Basic syntax → variable = input("message")

Ex:- name = input("name")

Print ("Hello, name")

→ Type of conversion while taking input

• Since input is always string, so 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 separate statements

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

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

Using split()

a, b = map(int, input("Enter two numbers:"))

Print(a, b)

→ Formatted output can be followed as :-

- Using f-strings for

```
name = "Riya"
```

```
age = 24
```

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

- Input & output with expressions

```
a = int(input("Enter number:"))
```

```
print("Square:", a*a)
```

- Custom separator & end

```
print("html", "css", sep=",")
```

```
print("Hello", end=" ")
```

```
print("mine")
```

(4) Discuss Control statements and Decision-making statement in python.

→ Control statements

- Control statements are used to control the flow of execution of a python program.
- They decide which statement is executed, how many times executed & stops; can write decision making, looping.

→ Types of control statements

- They are classified into 3

→ Decision making statements

→ Looping statements

→ Jump/control transfer "

→ Decision making statements are used to execute code based on conditions.

i) if statement

- Execute block of code only if the condition is true
- if it is false block skips.

Syntax

```
if condition:  
    Statement
```

Ex:-

```
age=17  
if age >=18
```

```
print("Eligible to vote")
```

ii) if-else statement

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

Syntax:-

```
if condition:  
    Statement  
else:  
    Statement
```

Ex:-

```
num=5  
if num%2==0  
    print("Even number")  
else  
    print("Odd number")
```

iii) if-elif-else statements

- Used to check multiple conditions
- elif means "else if".

Syntax:-

```
if condition:  
    Statement  
elif condition(2):  
    Statement  
elif condition(3):  
    Statement  
else:  
    Statement
```

Ex:-

```
num=9  
if num>0:  
    print("Positive number")  
elif num<0:  
    print("Negative number")  
else:  
    print("zero")
```

(3) Logical operators

- logical operators used to combine multiple conditions, used in decision-making statements

Operator

and

→ True if both are true

or

→ True if anyone is true

not

→ reverse the result

(4) Assignment Operators

- These operators are used to assign or update value in a variable & combine arithmetic operation.

Operator

Ex

=

$z=10$

$i = z/2$

$+=$

$z+=3$

$-=$

$z-=2$

$*=$

$z*=2$

(5) Membership Operators

- This operators check whether a value exist in a sequence , used with lists, strings, tuples.

Operator

in

present

not in

not present

(6) Identity operators

- Identity operators check whether 2 variables refer to same memory object , not compare values.

Operator

is

same object

is not

Different object

(b) Write an essay on Python programming fundamentals.

→ Role of Programming in Problem Solving

Programming helps in automating repetitive tasks. It involves improving accuracy and saving time. It involves breaking down a complex problem into smaller steps and perform each step logically.

→ The simplicity allows programmers to focus more on problem solving than syntax rules. Few lines compared to other languages.

→ Comments are used in python to explain code.

They help programmers document the purpose of variable, functions and logic used in the program. It makes readable & understandable.

→ Data types, operators and input/output operations to store data. Perform operations like arithmetic, relational & logical operations. Input / output operations allow interaction with user.

→ Decision making statements as if, else, elif and control flow statements. Order program statements are executed. Decision making help in implementing different actions.

Python helpful language for beginners and professionals. Python solve problems logically.

Real-world problems using python

(1) Movie ticket pricing

A movie theatre charges:
 ₹150 - children (age < 13)
 ₹250 - adults (age 13-59)
 ₹200 - Seniors (age ≥ 60)

*if a person watching
3D movie . extra ₹50

→ # Movie ticket pricing

```
age = int(input("Age"))
is3D = int(input("3D"))
if is3D == 1:
    price += 50
if age < 13:
    price = 150
elif age < 60:
    price = 250
else:
    price = 200
print(price)
```

(2) College attendance rule

A student is allowed to write exam if : attendance ≥ 75

OR attendance ≥ 60

AND has medical certificate (1=yes, 0=no)

Take attendance & medical certificate as input/output
"Allowed" or "not allowed".

→ # college attendance

```
attendance = int(input("Attend"))
medical = int(input("med"))
```

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

print("Allowed")

else: print('not 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 customer is a prime member, get extra 50% discount

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

Print final amount to be paid

$\rightarrow \text{bill} = \text{float}(\text{input}("Bill"))$

$\text{prime} = \text{int}(\text{input}("Prime"))$

if $\text{bill} \geq 5000$:

 discount = 20

elif $\text{bill} \geq 2000$:

 discount = 10

else:

 discount = 0

if $\text{prime} == 1$:

 discount += 5

~~float total = bill - (bill * discount / 100)~~

print(total)

(4) Smartphone battery warning

A phone shows: "Low battery" if battery ≤ 20

"Normal" if battery b/w 21-80

"Full" if battery > 80

But "phone is charging, it should show "charging" instead of any message. Input: battery percentage, is charging (1 or 0)

$\rightarrow \text{battery} = \text{int}(\text{input}("batl"))$

$\text{charging} = \text{int}(\text{input}("char"))$

if $\text{charging} == 1$:

 print("charging")

elif $\text{battery} \leq 20$:

 print("Low Battery")

elif $\text{battery} \leq 80$:

 print("Normal")

if $\text{charging} == 1$ else:

 print("Full")

(5) Driving License check

A person can get a driving licence if:
 $\text{age} \geq 18$

AND passed driving test ($1 = \text{yes}$)

But if $\text{age} \geq 60$, driving test is not required

Input: age, test passed . Print "Eligible" or "Not eligible"

```

→ age = int(input("Age "))
    test = int(input("Test "))
if age >= 18 and test == 1:
    print("Eligible")
elif age >= 60:
    print("Eligible")
else:
    print("Not 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, isGold (1 or 0), distance.

Print "free delivery" or "Delivery charged".

```
→ amount = int(input("amt "))
```

```
isGold = int(input("Gold"))
```

```
distance = int(input("Dist"))
```

if distance > 10:

print ("Delivery charged")

elif amount >= 500 or isGold == 1:

print ("free delivery")

else:

print ("Delivery charged")

(7) Bank loan approval

A bank approves a loan if:

Salary $\geq 30,000$ AND Credit Score ≥ 700

OR

Salary $\geq 50,000$ (Credit score ignored)

Input: salary, credit score . Print "loan approved" or "loan Rejected"

```
→ salary = int(input("Sal"))
    credit = int(input("Credit"))
    if (salary >= 30000 and credit >= 700) or salary >= 50000:
        print("Loan Approved")
    else:
        print("Loan Rejected")
```

(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

```
→ units = int(input("Units"))
if units <= 100:
    bill = units * 2
elif units <= 200:
    bill = units * 2 + (units - 100) * 3
else:
    bill = 100 * 2 + 100 * 3 + (units - 200) * 5
print(bill)
```

(9) A student's scholarship

A student gets a scholarship if:

mark ≥ 85 AND family income < 50000

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

```
→ marks = int(input("Marks"))
    income = int(input("Income"))
    single = int(input("Single"))
    if marks >= 85 and income < 50000 or single == 1:
        print("Scholarship Granted")
    else:
        print("Not Eligible")
```

(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]

→ theory = int(input("theory"))

pract = int(input("Pract"))

if (theory ≥ 40 and pract ≥ 40) or theory + practical ≥ 100:
 print("Pass")

else:

 print("Fail")

(11) Hotel room pricing

A hotel charges:

₹3000 per day for normal days.

₹4000 per day on weekends

If customer stays more than 3 days, gives 15% discount

Print final_bill

isWeekend = int(input("week"))

days = int(input("Days"))

if isWeekend == 1:

 Cost = 4000

else:

 Cost = 3000

total = Cost * days

if days > 3:

 total = total * 0.15

Print (total)

(12) Gaming level unlock

A game unlocks next level if:

score ≥ 100 OR player has a premium pass

But if player used cheating, access is denied

Input : score, isPremium, usedcheat

Score = int(input("score"))

Premium = int(input("Premi"))

```
cheat = int(input("cheat"))
if cheat == 1:
    print("Access Denied")
elif score >= 100 or premium == 1:
    print("Level Unlocked")
else:
    print("Level Locked")
```

(3) 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: datause, hasUnlimitPlan, roaming

```
data = float(input("Data"))
unlimited = int(input("Unlimit"))
roaming = int(input("Roam"))
if roaming == 1:
    print("Limited Data")
elif data <= 2 or unlimited == 1:
    print("Unlimited Data")
else:
    print("limited Data")
```

(4) Office Entry system

An employee can enter the office if:

ID card is valid AND

(fingerprint matches OR facescan matches)

But if it is a holiday, entry is denied for everyone

```
id = int(input("ID"))
finger = int(input("finger"))
face = int(input("face"))
holiday = int(input("Holiday"))
```

```

if holiday ==1:
    print ("Entry denied")
elif id ==1 and (finger==1 or face==1):
    print ("Entry Allowed")
else:
    print ("Entry Denied")

```

(5) Movie Rating Display

A movie app shows rating based on average score:

Average $\geq 8.5 \rightarrow$ "Excellent"

Average b/w 6.0 and 8.4 \rightarrow "Good"

Average $< 6.0 \rightarrow$ "Average"

But if the movie is marked as editor's choice, always show "Recommended".

Print the message.

```
rating = float(input("Rating"))
```

```
editor = int(input("Editor"))
```

```
if editor ==1:
```

```
    print ("Recommended")
```

```
elif rating  $\geq 8.5$ :
```

```
    print ("Excellent")
```

```
elif rating  $\geq 6$ :
```

```
    print ("Good")
```

```
else:
```

```
    print ("Average")
```