

1. Explain programming and python in detail

a) Definition and purpose of programming

Programming is the process of giving instructions to the computer so that it can perform tasks and solve complex problems.

The Python is a popular and most basic language of computer programming. It is easy to read and can handle all tasks easily, without any difficulties.

b) Characteristics of Python

i) object oriented language

ii) High level language

iii) Easy to learn & maintain

iv) Interpreted language

Applications of Python

i) Machine learning

ii) Gaming and graphics

iii) Web development

iv) Data science

v) Internet of Things

① ~~What are comments~~

Types of comments

1) Single line comment (//) -
2) Multi line comment
3) * Single line comment :- Single line
comment // is used for brief explanation
and are created by using the hash //
symbol // is used for explaining
what the Multi line comment These comments
are recommended way for block. this are
used to play with a hash // symbol at
the beginning of each line.

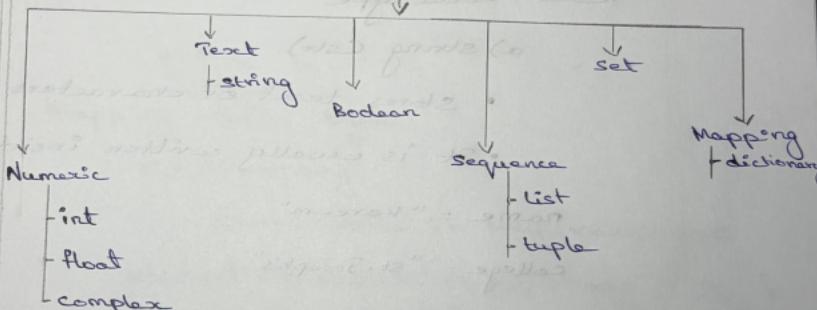
Importance of python in modern
Software development

- 1) Large number of built-in libraries
- 2) Works on windows, macos
- 3) Highly used in data science
- 4) Strong community support
- 5) Easy to maintain
- 6) Easy to scale
- (7) Widely accepted in software industry.

②

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

Data type specifies what kind of data a variable holds and Python determines data automatically by using the type() function.



Numeric data type

a) Int

* Stores whole number

* Positive or negative

$$a = 10$$

$$b = -5$$

b) float

* Stores decimal number

(square root of 25)

working with objects $x = 6.24$ off class named B

$$y = 3.28$$
 off class named A

$$z = 5.63$$

also for last value suppose off class

c) complex

both variables having two class addition &
stores numbers in obj format
nothing is kept all pieces just place values
 $sc = 3 + 6j$
keep data to methods

Reset data type

a) string (str)

- Stores text or characters

• It is usually written inside quotes

name = "Kareem"

college = "St. Joseph's"

Boolean data type

stores only true or false and is used in decision making.

is_Valid = True

is_Valid = False

Sequence data type

a) List -

collection of ordered

elements (number arrays)

Mutable (can change)

(3)

lists are b) tuples (ordered) structures

size is constant ordered collection

Immutable (can't be changed)

colors = ("Blue", "Black", "white")

set - data type means values

unordered collection and has duplicate

Value of sets

01 (dis) turing s = {a, b, c} true

1- (dis) turing

02 - (dis) Mapping data type

a) dictionary form

stores data in key value pair

example student = {"name": "Abdul", "age": "20"}

python operators

operators are special symbols

used to perform specific operations on
multiple values

one or more operands

1) Unary operator: operation that require
one operational to perform a specific
operation are known as unary operator

2) Binary operator: operations that require
two operands to perform a specific operation

Operands Variables / values that are used with the operator to perform a specific operation or task statement

Cat 1. Arithmetic operator

operator meaning of example

Addition sum of two numbers $a=5$

- Subtraction $b=6$ $a-b$

* Multiplication $a \times b$ $print(a+b) = 10$

/ Division a/b $print(a-b) = -1$

% Modulus division $a \% b$ $print(a \times b) = 30$

2.) Comparison operator

operator meaning of example

$=$ Equal to $x = 5$

\neq not equal to $y = 20$

$>$ greater than $print(x > y)$

$<$ less than

\geq greater/equal to

\leq less or equal to

values positive so result also positive

values negative so result also negative

values null so result also null

values infinity so result also infinity

(4)

3. Logical operator matching method

operator meaning I/o to Example

and if both are true $x > 50$

or if any one is true $y > 60$

not reverse the result print(x & y
and y > 55)

logical &

(("x & y") logical and

(("x & y"), "and") logical

print x & y; flag #

("x & y") logical address not

("x & y") logical

<"x & y"> #

tags present was addressed after

((("x & y"), "and")) logical tag up ←

Logical address

((("x & y"), "and")) logical tag - seq ←

((("x & y"), "and")) logical

logical tag

((("x & y"), "and")) logical

3. Explain python input and output operation
in detail programs

Input function: It is used to take user input and it always gives a string type in return

```
#input
name = input ("Kareem")
print ("Welcome, " + name)

# default type is string
roll_number = input ("94")
print ("94")

# class 'str'>
```

Type conversion while taking input

```
→ age = int(input ("Enter age : " "20"))
print (type(age))

→ price = float(input ("Enter price : 500"))
print (type(500))

# output
```

```
input ("Enter true or false : True")
is_valid = answer.lower () == "true"
```

(5)

Taking multiple inputs

- Using "split()

a, b = input("Enter two numbers : 3,6").split()

and now we are going to use split()

print(f" a = {3} , b = {6}")

- With type conversion

x, y = map(int, input("Enter two integers : -5, -6").split())

print(f" sum = {x + y}")

- Taking list input

numbers = list(map(int, input("Enter numbers : 4, 8").split()))

print(f" sum = {x + y}")

print(f" Numbers : {numbers}")

The print() function and formatted output

* # basic printing

print("Hello, world")

* # printing variables

name = "Kareem"

age = "20"

print(name, age)

* # customizing separator of print

```
print("python", "is", "good", sep="-")
```

output python-is-good

* # printing on the same line

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

```
print("world")
```

output Hello world

```
("{} + {} = {})".format
```

Logs is used for

({}) logging.info("message")

({}) logging.error("error")

```
("{} plus {} = {})".format
```

```
("product : result", "x", "y")
```

better not use writing directly

writing instead to do

```
("{} * {} = {}".format
```

selected printing it +

"result" = result

"{}" = op

(6)

A 4/1 Discuss control statements and decision making statements in Python.

control statements let you decide what part of your code runs. They allow programmer make decisions, take actions etc

Importance

- 1) Helps in decision making
- 2) Helps in allowing the repetition of code blocks
- 3) Dynamic and interactive program making
- 4) Low code redundancy

Types of control statements

- 1) conditional / decision making
- 2) looping statement
- 3) Jumping statement

* If it's only if the condition is true

Syntax about loops at lesson 10

example if condition else if condition not

: (2. # code or not

(3. loop

*) If Else :- Executes only one block of code if the condition is true

Syntax :- if statement looks

just like if condition is true then

elif, else # code if true or going to else

Else : # code in else

code in false

f) if - elif - else elif (

multiple sets of multiple condition checked top

to bottom should also go

Syntax is like example 6

if condition 1: # some code

else # code block else (

elif condition 2: # some code

else # code block else (

elif: # some code

default code block else (

2.) Looping statements :- It is used when

you want to repeat tasks

for loop :- Iterates over a sequence

for i in range(3, 6):

print (i)

While loop :- Runs until a condition becomes false

count = 0

while count <= 5 :

 print (count)

 count += 1

3.) Jump / loop control statements :- They help in interrupting or modify the loops
Break :- helps in stopping the loop

completely

for i in range (50):

 if i == 10

 break

 print (i)

Continue : This skips the current iteration

for i in range (5):

 if i == 2

 continue

 print (i)

it does nothing
if $x > 0$: else command

passes true

else passes false

(true) true

false

it is translates into as just first (c)
and fibres so first will be just
and all passes in else : stand

completely

:(2) pass in is not

at = 1 fi

stand

(1) true

now with 29-12 2nd : condition

no break

:(2) pass in is not

at = 1 fi

(1) S. Write an essay on Python programming fundamentals as used today.
Topic: Introduction.

Python is a high-level

interpreted programming language that has gained large popularity because of its very easy to learn and flexible to use. The Python programming is suitable for beginners as well as the experienced programmers.

Role of programming in problem solving

programming allows humans to communicate instructions to a system in a logical manner. While solving a problem a programmer first analyzes the requirement such as, planning the steps, writing the code and finally checking the output. Use of comments for code documentation comments plays an important role in planning the functionality. comments

can be written by using the hash (#) symbol for one or more quotes

Data type operator and Input Output
operations

and lastly Python provides several built-in
such as numbers as strings, lists, tuples
sets and boolean. operators are used to
perform calculations and logical comparisons

User inputs are handled by using input() function
and output using print() for
better clarity

use of comments for code documentation

comments play an important role in

Explaining code functionality. In python
comments can be written using the
hash (#) symbol

Conclusion: The python programming focuses
on clarity, simplicity and logical program for
problem-solving. The strong control structures
makes it a valuable language for both

learning and professional use

(@)

program 1) Movie ticket pricing

```
age = int(input("Enter age"))
if age <= 30:
    print("Is it a 3D movie?")
    choice = input("(1 for Yes, 0 for no):")
    if choice == "1":
        price = 150
    else:
        price = 100
else:
    price = 200
```

elif

(age <= 60)

base price = 250

else:
 price = 900

if is_3D == "Yes":

price += 200

print("Ticket price : ", price)

0000 -> 0001 file

01 - 10000 file

1 - 10000 file

2 - 10000 file

called at 10000 (0001) file

program 2) college attendance

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

a = float(input("Attendance :"))

m = int(int(input("Medical (%):")))

if a >= 75 or (a >= 60 and m == 1)

print("Allowed")

else

print("not allowed")

program 3) E-commerce discount

bill = float(input("Enter bill amt :"))

prime = int(input("prime member (%):"))

discount = 0

if bill >= 5000

discount = 90

elif bill >= 2000

discount = 10

if prime == 1

discount += 5

print("final amount to be paid :")

program 4) smart phone battery warning

if plugged == 1

status = "charging"

else battery level + 10

if level <= 20 = beep

(("warning") to status = "low battery")

elif level <= 80 : say

status = "Normal"

else

status = "full"

print (status)

program 5) driving license check

Eligible = false

if years >= 60 : Eligible

Eligible = true

Eligible = true

Eligible = true

print ("Eligible")

else

Print ("not eligible")

program 6) Online food delivery

distance > 10km = never free

order amount > 500 | gold member

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

gold = int(input("gold member:"))

km = float(input("distance:"))

free = False

if km > 10:

 if amount >= 500 or gold == 1:

 free = True

 if free:

 print("free delivery")

else:

 print("Delivery charge")

program 7) Bank loan Approval

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

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

if salary >= 30,000 and credit score >= 700
 or salary >= 50,000:

 print("loan approved")

else (Candidate's loan)

 print("loan rejected")

(Candidate's loan) print

⑪

program 8) Electricity bill (using if)

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

if units <= 100

bill = units * 5

else if units <= 200

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

else

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

print ("Bill amount : ", bill)

program 9) Student scholarship

marks = int(input("Marks :"))

income = int(input("Income :"))

single_parent = int(input("Single parent :"))

if marks >= 85

if single_parent = 1

print ("scholarship")

elif income <= 5,00,000

print ("scholarship")

else

print ("no scholarship")

else

print ("scholarship")

(1)

program 10) // Exam result

```
( theory = int(input("Enter marks of theory:"))
practical = int(input("Enter marks of practical:"))
total = theory + practical
if (theory >= 40 and practical >= 40) or
    (total - theory <= 10 and total >= 100):
    print("pass")
else:
    print("fail")
```

program 11) Hotel Room Pricing

```
weekend = int(input("Enter weekend stay:"))
dayspent = int(input("Enter days spent:"))
if weekend == 1:
    bill = dayspent * 4000
else:
    bill = dayspent * 3000
if dayspent > 3:
    bill = bill - (bill * 15 / 100)
print(bill)
```

(get calculate room bill)

ad 7

(get calculate room bill)

program 12) Game level unlock

```

score = int(input())
is_premium = int(input())
used_cheat = int(input())

if used_cheat == 1
    print("Access Denied")
elif score > 100 or is_premium == 1
    print("level unlocked")
else
    print("level locked")

```

program 13) Mobile data usage

```

data_usage = float(input())
is_roaming = int(input())
has_unlimited_plan = int(input())

if is_roaming == 1
    print("unlimited not allowed")
elif data_usage <= 2 or has_unlimited_plan == 1
    print("unlimited access")
else
    print("limited access")

```

Program 14) Office Entry System

```
isValid = int(input("))  
fingerprint = int(input("))  
facescan = int(input("))  
holiday = int(input("))  
if holiday == 1  
    print("entry denied")  
elif isValid == 1 and (fingerprint == 1 or  
                      facescan == 1)  
    print("entry allowed")  
else  
    print("entry denied")
```

Program 15) Movie Rating Display

```
average rating = float(input("))  
editor choice = int(input("))  
if editor choice == 1  
    print("Recommended")  
elif average rating >= 85:  
    print("Excellent")  
elif average rating >= 60:  
    print("Good")  
else  
    print("Average")
```