Assignment

1. Explain programming and python in detail

Def :~ Programming is the process of writing instructions to Computer follow to perform particular task

Purpose :~ The main purpose of programming is to Communicate with System to solve problems and automate task by writing step by step instructions

Python def :- Pythan is a high-Level language it is interpreted and object oriented help for developers in developing web applications Software development etc.

Characteristics and applications of python

-> Easy to learn and Read
-> platform independent
-> object oriented programming
-> Interpreted language
-> web development
-> Software development
-> Game development
->Automating and Scripting

Types of Comments

There are two type of Comments in python

1) Single line Comment (#)

Ex) # Just. Hello program

Print ("Hello")

2 Multiline Comment (""" """)

Ex) """This is
Multiline
Comment"""

Importance

In modern Software development python will play a key role due to it's Simplicity and understanding and due to it vast libraries which are more important. This reduce the difficulty in the Software development.

2) datatypes and Operators in python with example

Built-in Data types

Numeric → int, float, Complex

integer values, float values, Complex values

Sequence → str, list-type

String → immutable Sequences

list → Ordered, mutable [ ]

types → Ordered immutable. ( )

Set → set. frozenset

↓ ↓

Mutable immutable

{ }

Mapping → dict → unordered key-value pairs

Boolean → Bool → Represents True/False

Types Identification

Type ( ) → determine the type of variable

Ex :- x = 10

Y = "hello"

Print (type (x))

Print (type (y))

Various python Operator

Arithmetic Operator

"+" addition → Ex : a+b add 2 operands

"-" Substraction → Ex: a-b Substract right operand from left
"*" Multiplication → Ex: a*b . Multiplies 2 operands
"/" division → Ex: a/b → divide left to right operand
"//" floor division → a//b → divide and return quotient
"%" Modules → Retun remainder Ex: a % b
"**" Exponential → a**2 → return power

Assignment

"=" x = 5
"+=" x += 5
"-=" x -= 5
"*=" x += 2
"/=" x /= 2

Comparison operator

== Equal 5 == ?
!= Not equal 5 != 2
> greaterthan 5 > 2
< Lessthan 5 < 2
>= greater than equal 5 >= 2
<= Less than equal 5 <= 2

Logical operator

and Return True if both statement are True
or Return True if one of statement is true
not Reverse the Result

Membership Operater

in Return True if a value is present in Sequence
not in Return True if a value is not present in Sequence

Identity operators
is Return true if both Variables point Same object
is not Return true if both Variable are not point to Same object

Real-world usage of Operators
Arithmetic operators → game development
Comparison / Logical, Membership

3) python Input / output Operations
Input ( ) → helps to provide input
+ always return in string
input ("Enter name")

Type Conversion
+ helps to Convert Type of Variable
* as input ( ) is string type can be Convert it int int, float etc...
Ex:- age = int (Input ("Enter age")

Taking multiple Inputs
* There are Several ways to take multiple inputs on Single line using Split ( ): used to split a Single input string into a list of strings
* Multiple elements can be splited.
using Map ( ) with split ( ): To Convert multiple inputs to specific types immediately
Ex: a, b = Map (int (input ("Enter 2 integers:"). Split ( ))

Formatted output using print ( )
Print ( ) → Sends Output to Console
Separator (sep) :- Separted multiple arguments with a Sugle Space

Print ('apple', 'banana', 'cherry')
Print ('apple', 'banana', 'cherry', Sep = '|')
-Fstrins :- you Can Control precision and alignment using a colon: within the braces
-format ()
Print ("Hello. {}. You are {}".format ("Bob", 21))

4. Controle statement and decision making statements meaning & importance of Control Statements
Control statement are Crucial because they allow programmers to give Condition/instructions to be followed in Code like decision making, iteration flow, Controle
Type of Controle statements
if, elif, use → Execute base on Specific Condition for,
while → Execute block of Code repeatedes.
break, Continue, pass, return → flow within loops
decision - making statement
if statement
* simplest form, executing block of Code only if True if Condition
    # Code to execute
if - else statement
    + True Condition is executed if True
    if Condition:
        # Code block for True
    else: # Code block for false

like if - elif - else statement

* multiple condition sequentially

```
if Condition1:
    # Code block for Condition True
elif Condition:
    # Code block for Condition True
else:
    # Code block if all Condition are false
```

3) Write an essay on python programming fundamentals

Role of programming in problem-Solving

* programming is essentially the process of translating a Problems Solution into sequence of instruction a Computer can execute
* help for breaking down Complex issues into Smaller, Steps

Syntax Simplicity and Readability

* Python is clear, readable Syntax in English

```
fruits = ["apple", "banana", "cherry"]
for fruits in fruites
    Print (f" I like {fruit}")
```

Use of Comments for Code documentation

Comment like Single and multiple line are useful for code

* # for Single line
* ''' ''' for multiple line

data type, Operators, I/O operations

int, float, str, bool → data types

Operator are Several types

like arithmetic ; logical. Relations. - Assignment, membership identity.

I/O operations

for input ( ) ==> Enter default string

output -> print ( )

Control flow using decision - Making statements

* The Flow Controlled by Control Statements

if . else, elif

Ex :- age = 20

```
if age >= 18
    Print ("eligible")
elif age >= 13:
    Print ("not")
else:
    print ("child")
```

Python programming

1) Movie Ticket pricing

```
def ticket (age, is3D):
    if
        age < 13:
        Price = 150
    elif age >= 13 and age < 59:
        Price : 250
    else:
        Price = 20
if is3D == 1:
ticket_price = price
```

```
def ticket_price
age = int(input("Enter age:"))
is_3D = int(input("is 3d:"))
final_price = ticket(age, is_3D)
print(f"The final price {final_price} Rs")
```

2) College Attendance Rule

```
attendance = float(input("Enter attendence percentage"))
Medical_Certificate = int(input("Do you have medical
    Certifical {(1for yes 0 for No):"))
if attendance >= 75:
    Print("Allowed")
    elif attendance >=60 and medical Certificate == 1:
        Print("Allowed")
else:
    Print("not Allowed")
```

3) E-Commerce discount

```
def final_amount(bill_amount, is prime):
    dis = 0 # discount
if bill_amount > 500:
    dis = 20
elif 2000 <= bill_amount <= 4999
    dis = 10
else:
    dis = 0
if is prime == 1:
    dis += 5
```

(6) (7)

dis t = 5
discount = (dis/100) * bill-amount.
final amount = bill- amount - discount
return final amount
bill = float (input ("Enter bill amount:"))
Prime = int (input (" is prime".))
result = final - amount (bill, prime)
Prime ("final amount + paid: ", result)

Smartphone Battery warning

```
def message (battery, ischarging):
  if ischarging == 1:
      return "charging"
  elif battery <= 20:
      return "Low Battery"
  elif 21<= battery <= 80:
      return "Normal"
  else return " Full"
Print (message (15, 0))
```

Driving License check

```
age = int (input ("Enter age :"))
testpassed = int (input ("did they Pass"))
  if age >= 60
    Print ("Eligible")
  elif
```

```
# Input Inputs
age = int (input ("Enter age :"))
test passed = int (input ("Did they pass the test ? (1 = yes,
                          0 = no) :"))
# Evaluation
if age >= 60:
    Print ("Eligible")
elif age >= 18 and test passed == 1 :
    Print ("Eligible")
else :
    Print ("Not Eligible")
```

Online Food delivery

```
def : check_free_delivery (amount, is Gold, distance):
# Condition 1 : Distance is too far, delivery is never free.
if distance > 10:
 return False
# Condition 2 : Order amount >= 500 or user is a gold
 member (will elif amount >= 500 or is Gold == 1 :
  return True
 else :
   return False.
Print (f "Amount : 200, Gold : 1, Distance : 5 -> Free :
        {check_free_delivery}
```

Bank Loan Approval

```
# Input : Salary and Credit Score
Salary = float (input ("Enter Salary :"))
Credit_Score = int (input ("Enter Credit Score :"))
# Loan Approval Logic
```

```
if (salary >= 30000 and Credit_score >= 700) or (salary >=
    50000):
    Print ("Loan Approved")
else:
    Print ("Loan Rejected")
```

Electricity Bill unit Consumed

```
unit_Consumed = 250
bill_amount = 0
if units Consumed <= 100:
    bill amount = units_Consumed * 2
elif units_Consumed <= 200:
    bill_amount = (100 * 2) + (units_Consumed - 100) * 3)
else:
    bill_amount = (100 * 2) + (100 * 3) + (units_Consumed - 200)
Print (f " Final bill amount: {bill_amount}")
```

Student Scholarship program

```
## input data
try:
    Marks = float (input (" Enter student Marks :"))
    income = float (input (" Enter family income : "))
## 1 for yes, 0 for NO
is_Single_parent = int (input ("Is Student a Single Parent
    child")
if Marks >= 85:
if is_Single_parent == 1 or income < 500000:
    Print ("Congratulations! Student is eligible for the
    Score
else:
    Print ("Sorry, Student is not eligible (Income too high")
```

```
else:
    Print ("sorry, Student is not eligible (Marks too low).")
except ValueError:
    Print ("Invalid input. please enter numerical values.")
```

9. Online exam Result

```
try:
    # Taking input from the User
    theory = float (input ("Enter theory Marks: "))
    Practical = float (input ("Enter practical Marks: "))
    total = theory + practical
# checking Conditions
    if total >= 100:
        Print ("Result: Pass (Total Marks >= 100)")
    elif theory >= 40 and practical >= 40:
        Print ("Result: PASS (Individual Marks >= 40)")
    else:
        Print ("Result: FAIL")
except ValueError:
    Print ("Invalid input. please enter numerical values.")
```

11. Hotel Room Bucing

```
def calculate_hotel_bill (is_weekend, days_stayed):
weekday_rate: 3000
weekend_rate = 4000
discount_threshold = 3
discount_rate = 0.15 # 15%
if is_weekend == 1:
```

```
    daily_rate = weekday_rate
total_cost = days_stayed * daily_rate
if days_stayed > discount_threshold:
    discount_amount = total_cost * discount_rate
    final_bill = total_cost - discount_amount
else:
    final_bill = total_cost
return final bill
print(f"Bill for 2 normal days: {calculate_hotel_bill(0,2)}")
```

Gaming Level unlock

```
score = int(input("Enter Score:"))
is_premium = input("Has premium pass? (yes/no): ").lower()
used_cheat = input("used cheat? (yes/no):").lower()
if used_cheat == "yes":
    print("Access Denied")
elif score >= 100 or is_premium == "yes":
    print("Next Level unlocked")
else:
    print("Level Locked")
```

Mobile Data Usage

```
def check_unlimited_data(data_used, has_unlimited_plan,
    is_roaming):
    if is_roaming:
        return "standard Data
        (Roaming)"
```

```
if data_used<=2 87
has_unlimited_plan:
    return "unlimited Data"
else:
    return "standard Data"
# Example
Print (check_unlimited_data (1.5, 0, 0))
# unlimited Data
```

Office Entry System

```
def can enter_office (id_valid,
 fingerprint, face_Scan, is_holiday):
    if is_holiday:
        return "Entry Denied (Holiday)"
    if id_valid and (fingerprint or face_Scan):
        return "Entry Granted"
    else:
        return "Entry Denied
Print (can_enter_office (1, 0, 1, 0)) ##
  Entry Granted.
```

Movie Rating Display

```
def get_movie_rating (average_rating,
    is_editors_choice):
    if is_editors_choice:
        return "Recommended"
```

1.

```
if average_rating >= 8.5:
    return "Excellent"
elif 6.0 <= average_rating <= 8.4:
    return "Good"
else:
    return "Average"
Print (Get_movie_rating (7.5, 1))
```