

Data

Analysis

Using

Python

Assignment

By

A. Kumari

3rd BSc Computer
Science

Admno: 19760

1. Explain Programming and Python in detail.

Programming: It is the process of designing, writing, testing and maintaining a set of instructions that a computer follows to perform specific tasks.

Purpose of programming:

- * To solve real world problems logically
- * To automate repetitive tasks
- * To process and analyze data
- * To control hardware and machines.

Python Programming language:

- * Python is a high-level, interpreted, general-purpose programming language.
- * It is created by Guido van Rossum in 1991.

Characteristics of Python:

- Simple and easy to learn :- which will be executable in fewer lines.
- Interpreted language: No need for compilation
- Object Oriented: It supports classes and objects
- Portable: runs on any windows.

Applications of Python:

- Web development
- Data Science and Data Analysis
- Desktop applications.
- AI and Machine learning.

Types of Comments in Python with syntax:

- Comments are used to explain code and improve readability.

They are ignored by interpreter.

* Single line comment: Used for short explanation

Syntax: # This is Single line comment code

* Multi line comment: It is used for long explanation. Python uses the triple quotes.

Syntax: """ This is multi line comment

It has several line of explanation

code

* In line Comment: It will written on the same line as code

Syntax: x=10 # Assigning a value.

Importance of Python in Modern Software development:

- * Faster development - less time consumed
- * High demand in industry - like google, social media website
- * strong community support.
- * Integration capability.

Q. Describe data types and operators in python with suitable examples.

In python, data types define the type of data variable can store.

while operators are symbols used to perform operations on variables and values.

Built in data types in python:

Python have several built in data types.

1. Numeric Data type:

Type	Description
int	integer value
float	Decimal value
complex	Complex numbers

Ex: a=10 # int

b=3.14 # float

c=2+3j # complex

2. Sequence Data type: To store ordered collection data

Type	Description
list	Mutable Sequence
tuple	Immutable Sequence
string	Sequence of characters

Ex: `list1 = [1, 2, 3]`

`tuple = (4, 5, 6)`

`name = "Python"`

3. Set Data type: Used to store unordered and unique elements

Ex: `set1 = {1, 2, 3, 3}`

`Print (set1)`

4. Mapping Data type: Used to store data in key-value pairs.

Ex: `Student = { "name": "Akash", "age": 21 }`

5. Boolean Data type: Used to store True or False values

Ex: `x = True`

`y = False`

Types Identification using `type()` function:

The `type()` function is used to identify the data type of a variable.

Syntax: `type (variable)`

Ex: `a = 10`

`Print (type(a))`

`b = "Python"`

`Print (type(b))`

Various Python operators:

Operators are symbols used to perform operations

on variables

1. Arithmetic Operators:

used to perform mathematical operations

Operator

operation

+

Addition

-

Subtraction

*

Multiplication

/

Division

%

Modulus.

Ex: $a = 10$

$b = 3$

Print($a+b$)

Print($a-b$)

2. Assignment Operators:

Used to assign values to variables

operator	operation
$=$	Assignment
$+ =$	Addition assignment
$- =$	Subtraction assignment
$* =$	Multiplication assignment
$/ =$	Division assignment
$\% =$	Modulus assignment

Ex: $x = 10$

$x + 5$

Print(x)

3. Comparison Operators:

Used to compare values

operator	operation
$= =$	Equal
\neq	Not equal
$>$	Greater than
$<$	Less than

Ex: $a = 5$

$b = 10$

Print($a < b$)

4. logical operators:

Used to combine conditional statements

Operator	Meaning
and	logical AND
or	logical OR
not	logical NOT

Ex:- $a = \text{True}$

$b = \text{False}$

Print (a and b)

5. Membership operator:

Used to test whether a value is present in sequence or not

Operator	Meaning
in	Present
not in	not Present

Ex:- $\text{list } l = [1, 2, 3]$

Print (2 in lists)

6. Identity operators

Used to compare memory locations of object

Operator	Meaning
is	Same object
is not	Different object

Ex:- $a = 10$

$b = 10$

Print (a is b)

Real world usage of operators:

→ Used in billing systems and calculator

→ Used in decision making programs

→ Used in searching data

→ Used to update value.

3. Explain python input and output operation in detail.

Input and output (I/O) operations:

In python allow program to interact with

the user.

Input operations: This is used to take data from the user.

Output operations: This is used to display result to the user.

Python mainly uses the `input()` function for input and the `print()` function for output.

* `input()` function and its default data type:

* The `input()` function is used to take input from the user through the keyboard.

Syntax: `variable = input("message")`

Example: `name = input("enter your name")`

`print(name)`

The default data type of `input()` is string (`str`), even if

the user enters numbers.

* Type conversion while taking input:-

Since `input()` returns a string, we must convert it to the required data type using type conversion function.

Common Type conversion function.

* `int()` → converts to integer

* `float()` → converts to decimal number

* `str()` → converts to string

Ex:- 1) `age = int(input("enter your age:"))`

`print(age)`

2) `price = float(input("enter price"))`

`print(price)`

* Taking Multiple Inputs:
Python allows taking multiple inputs in a single line

using the `split()` method.
Ex:- `a,b=input('enter two numbers:').split()`

```
a = int(a)  
b = int(b)  
print(a+b)
```

* Formatted Output using `print()` Function:
The `print()` function is used to display output

on the screen.

→ Using `sep` parameter is used to separate multiple values.

The `sep` parameter is used to separate multiple values.

Ex: `Print(10,20,30, sep="")`

→ Using `end` Parameter(`end`)

The `end` parameter controls what is printed at the end.

Ex: `Print("Hello", end=" ")`

```
Print("world")
```

→ `format()` Method: creating the formatted strings

This is used to format the output.

Ex: `name="Python"`

`version=3.10`

`Print("language:{} Version:{}".format(name, version))`

→ f-String / modern method

Ex: `marks=85`

`Print(f"marks obtained:{marks}")`

Discuss control statements and Decision - making statement
in python.

In Python, control statements are used to control the flow of execution of a program. Normally, Python programs execute statements line by line from top to bottom. Control statements allow us to change this normal flow.

Meaning and importance of control statements

- > Control Statement decide which statement should be executed
- > How many times a block of code should run, when to stop

Importance:-

- > Helps in decision making
- > Reduces complexity of program.

-> Makes program flexible and dynamic.

Types of Control Statements in python:

Control Statements in python are broadly classified into:

1. Decision-Making Statements

2. Looping Statements

3. Jump Statements

① Decision-Making Statement:

Used to execute different blocks of code based on condition

a) if Statement:

The if statement executes a block of code only when the condition is true.

Syntax: if condition:
 Statement

b) if - else Statement:

The if - else statement provides an alternative block when the condition is false.

Syntax: if condition:

 Statement

 else:

 statements

c) if - elif - else Statement:

Used to check multiple conditions

Syntax: if conditions:

 Statement

 elif condition 2:

 statements

 else:

 statements

Looping statements:

Used to execute blocks of code repeatedly

a) for loop:

A for loop is used to iterate over a sequence such as a list, tuple, string or range.

Syntax: for variable in sequence:
statement.

b) while loop:

A while loop is used to repeat a block of code as long as a condition is True.

Syntax: while condition:
statement.

3. Jump statements:

Used to alter the flow inside loops

a) break statement: The break statement is used to terminate a loop immediately when a condition is satisfied.

Syntax - break

b) continue statements: The continue statement is used to skip the current iteration and move to the next iteration.

Syntax: continue

c) Pass statements:

The pass statement is used as a placeholder where a statement is required syntactically but no action is needed

Syntax: pass

5. Write an essay on python programming ~~fundamentals~~.

Introduction:

Python is one of the most popular and widely used Programming language in the world today. It is a high-level interpreted, general purpose language known for its simplicity and readability.

Role of Programming in problem solving:

Programming plays a crucial role in problem solving by breaking complex problems into smaller, manageable steps. A programmer analyzes the problem, designs an algorithm and implements it using a programming language. Python with its simple syntax allows programmers to focus more on solving problems.

Python Syntax Simplicity and Readability:

One of the major strengths of Python is its simple and readable syntax. Python uses English-like keywords and relies on indentation instead of braces, which makes the code easy to understand and maintain.

Uses of comments for code documentation: Comments are used to explain the code and improve readability. They help other programmers understand the logic of the program and are useful for future maintenance. Python supports single-line comments using # and multiline comments using triple quotes.

Data types, operators and input/output operations: Python provides various built-in data types such as integers, floating-point numbers, strings, lists, tuples, sets, dictionaries. Operators are used to perform operations

on data, including arithmetic, comparison, logical and assignment operations.

Control flow using Decision-Making statements

control flow statements control the execution of a program. Decision-making statements such as if, if-else, and if-elif-else allow the program to make decisions based on conditions. These statements are essential for implementing logic in real-world applications like grading systems and validation programs.

Real-world Problems using Python Programming:

1 Movie Movie Ticket Pricing:

```
age = int(input("enter age:"))
```

```
is_3D = int(input("Is it a 3D movie? (1 for yes, 0 for no):"))
```

```
Price = 0
```

```
if age < 13:
```

```
    Price = 150
```

```
elif 13 <= age <= 59:
```

```
    Price = 250
```

```
else:
```

```
    Price = 200
```

```
if is_3D == 1:
```

```
    Price += 50
```

```
print("Final Ticket price: ₹", Price)
```

College attendance rule

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

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

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

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

```
else:
    print("Not Allowed")
```

5 3. E-commerce Discount

```
bill = float(input("enter bill amount :"))
is_prime = int(input("Is prime member? (1=yes, 0=no):"))
discount = 0
if bill >= 5000:
    discount = 0.20
elif bill >= 2000:
    discount = 0.10
else:
    discount = 0
if is_prime == 1:
    discount += 0.05
final_amount = bill - (bill * discount)
print("final amount to be paid = $", final_amount)
```

4. Smart phone Battery warning

```
battery = int(input("enter battery percentage :"))
is_charging = int(input("Is the phone charging? (1=yes, 0=no):"))
if is_charging == 1:
    print("charging")
else:
    if battery <= 20:
        print("low Battery")
    elif battery <= 80:
        print("normal")
    else:
        print("full")
```

5. Driving license check

```
age = int(input("enter age:"))
test passed = int(input("passed driving test ? (1=yes, 0=no):"))

if age >= 60:
    if age >= 18:
        print("eligible")
    else:
        print("not eligible")

else:
    if age >= 18 and test passed == 1:
        print("eligible")
    else:
        print("not eligible")
```

6. Online food Delivery

```
amount = float(input("enter order amount:"))
is gold = int(input("is user a gold member? (1=yes, 0=no):"))

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

if distance > 10:
    print("Delivery charged")
else:
    if amount >= 500 or is gold == 1:
        print("free delivery")
    else:
        print("Delivery charged")
```

7. Bank loan Approval

```
salary = float(input("enter salary:"))
credit score = int(input("enter credit score:"))

if salary >= 50000:
    print("loan approved")
```

elif salary >= 30000 and creditscore >= 700:

 Print("Loan approved")

else:

 Print("Loan rejected")

8. Electricity Bill

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

if units <= 100:

 bill = units * 2

elif units <= 200:

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

Print("Final bill amount : \$", bill)

9. Student Scholarship

Marks = int(input("enter student's marks:"))

income = float(input("enter family income:"))

singleparent = int(input("enter single parent status (1=yes, 0=no):"))

Scholarship = (marks >= 85) and (singleparent == 0 or income < 1000)

if Scholarship:

 Print("The student is eligible for a scholarship")

else:

 Print("The student is not eligible for a scholarship")

10. Online Exam Result.

Theory = int(input("enter theory marks:"))

Practical = int(input("enter practical marks:"))

Total = Theory + Practical

if (Theory >= 40 and Practical >= 40) or Total >= 100:

 Print("Result: Pass")

else:

 Print("Result : Fail")

11. Hotel Room Pricing

is weekend = int(input("enter 1 for weekend, 0 for normal day:"))
daystayed = int(input("enter number of days stayed:"))
if isweekend == 1:
 rate = 6000
else:
 rate = 3000
total Bill = rate * daystayed
if daystayed > 3:
 total Bill = total Bill - (total Bill * 0.15)
 Print("Final Bill Amount: RS.", total Bill)
Print("Final Bill Amount: RS.", total Bill)

12. Gaming level unlock

Score = int(input("enter Score:"))
is Premium = int(input("enter 1 if premium pass, 0 otherwise:"))
Used cheat = int(input("enter 1 if cheating used, 0 otherwise:"))
if used cheat == 1:
 Print("Access denied")
elif Score >= 100 or is Premium == 1:
 Print("Next level unlocked")
else:
 Print("Level locked")

13. Mobile Data Usage

dataused = float(input("enter daily data used (in GB):"))
has unlimited plan = int(input("enter 1 if unlimited plan, 0 otherwise:"))
is Roaming = int(input("enter 1 if roaming is ON, 0 otherwise:"))
if is Roaming == 1:
 Print("Unlimited data not available")
elif dataused <= 2 or has unlimited plan == 1:
 Print("Unlimited data available")

else:

 print("limited data only")

14. Office Entry System

```
id valid = int(input("enter 1 if ID card is valid, 0 otherwise:"))
fingerprint = int(input("enter 1 if finger print matches, otherwise:"))
facescan = int(input("enter 1 if face scan matches, otherwise:"))
is Holiday = int(input("enter 1 if today is a holiday, 0 otherwise:"))

if is Holiday == 1:
    print("entry denied")
elif id valid == 1 and (fingerprint == 1 or facescan == 1):
    print("entry allowed")
else:
```

 print("entry denied")

15. Movie Rating Display

```
average Rating = float(input("enter average rating:"))
is Editor choice = int(input("enter 1 if editor's choice, 0 otherwise"))

if is Editor choice == 1:
    print("Recommend")
elif average Rating >= 8.5:
    print("excellent")
elif 6.0 <= average Rating <= 8.4:
    print("good")
else:
    print("Average")
```