S.Kousar Begum
III Biotechnology
Data analysis


1. Explain programming and python in detail.

Definition and purpose of programming

Definition: Programming is process of developing and writing a set of instructions using a programming language, so that a computer can perform specific tasks.

Purpose:

→ To solve problems using computers
→ To perform tasks automatically
→ To develop applications, software, websites and systems
→ To process data and make decisions
→ Improve speed and accuracy
→ process and analyse data (weather predictions, bussiness reports, data analysis)

Characteristics and applications of python characteristics:

Python has several important characteristics that make it popular and easy to use

→ Simple and easy to learn: Python has clear and englich like syntax, so beginners can understand it easily.
→ High-level language: programmers do not need to manage memory (or) hardware details python handles them automatically.
→ Interpreted language: python executes code line-by-line, which makes debugging easier.
→ Object oriented: Python supports classes and objects & concepts like inheritance and polymorphism.
→ Platform Independent: Python programs can run on windows, linux and mac without modification
→ Dynamically typed: Data types are assigned automatically at run time, no need to declare them.

→ large standard library: Python provides many built-in for tasks like math, file handling, networking etc.

Applications: Following are the applications of python:

→ Websites: used to make websites and web applications

→ Data analysis: used to study and understand large amount of data

→ machine learning and AI: used to make smart systems like face recognition etc.

→ Automation: used to do repeated tasks automatically

→ Games: used to make simple games

→ cyber security: used to make software for computers

→ Robotics and IOT: used in small devices and robots python is used for websites, data analysis, AI/ML automation, games apps, security and robotics.

Types of comments in python with syntax:

Python supports three types of comments

→ Single-line Comment: used to write short comment in one line and starts with the # symbol.

Syntax: # this is a single-line comment

→ Multi-line Comment: used to write long comments in multiple lines and written using triple quotes

''' ''' (or) """ """

syntax: '''
This is a
multi-line comment in python

→ In-line Comment: Comment written at end of a statement and used to explain specific part of code

Syntax: x = 10 # assigning value 10 to x
print(x) # printing the value of x

S. Kousar Begum
III Biotechnology
Data analysis


Importance of python in modern software development:
Python is very important in todays software world because of its simplicity, power and wide usage.
Some key points are:
→ Easy to learn
→ saves time
→ used in modern technologies
→ many libraries available
→ works on all platforms
→ used by Big companies
→ large community
→ works in many fields

② Describe data types and operators in python with suitable example.
→ Built in Data types in Python (Numeric, sequence, set, mapping Boolean)
Datatypes: A data type tells what kind of value is stored in a variable such as numbers, text, true/false etc.
Python has different data types:
int - Whole numbers (no decimal)
float - Decimal numbers
complex numbers with imaginary part (a+bj)

Ex:- x = 10 # int
y = 3.14 # float
z = 2+3j # complex

→ Sequence datatypes: These store multiple items in an ordered manner.
Common sequence types are:
String (str): Stores text, string data types are written in quotes. Ex:- name = "Python"

List : These are ordered, mutuable and allows duplic
are written in "[]"
Ex: fruits = ["apple", "banana", "mango"]

Tuple: These are ordered but not changeable. Tuple use "()"
Ex: ("red", "green", "blue")

Set Data types : stores unordered items and no duplicates are allowed. set data types uses "{}"
Ex: numbers = {1, 2, 3, 4}

Mapping Datatypes: stores data in key : Value pairs
It uses "{}" only dictionary belongs to mapping type
Ex: student = {"name": "Meena", "age": 20}
Here "name" and "age" are keys
"Meena" and 20 are values

Boolean Data types: Store True (or) false values
used in conditions and logic. Boolean often comes from comparision
Ex: print (10>5) # True
print (2==3) # false

Type identification using type()
In python you can identify the type of variable (or) value using the built in type () function.
It tells you what kind of objects it is (like int, str, list, etc)
Syntax: type (object)
Object → the variable (or) value you want to check
Returns the type of object
Example:
Numeric types:

S.Kousar Begum
III Biotechnology



```
print (type (x)) #<class 'int'>
print (type(y)) # < class 'float'>
```

→ String type:

```
name = "Meena"
print (type (name)) #<class 'str' >
```

→ Boolean type:

```
is_valid = True
print (type (is_valid)) #<class 'bool' >
```

→ list, Tuple, set, dictionary

```
my_list = [1,2,3]
my_tuple = (1,2,3)
my_set = {1,2,3}
my_dict = {'a' : 1, 'b' : 2}
print (type (my_list))  # < class 'list'>
print (type (my_tuple)) # < class 'tuple' >
print (type (my_set))   # < class 'set' >
print (type (my_dict)) # < class 'dict'>
```

check if variable belongs to certain type

```
Ex: x=10
    print (isintance (x, int)) # True
    print (isintance (x, float) # False
```

Various python operators (Arithmetic, Assignment, Comparision, logical, membership, identity)

Operator: An operator is a symbol that tells the computer to perform certain mathematical and logical operations. There are different operators in python:

→ Arithmetic Operators: These are used for doing some basic mathematical calculations.

| Operator | meaning | Example | Result |
|---|---|---|---|
| + | Addition | 10+5 | 15 |
| - | subtraction | 10-5 | 5 |
| * | Multiplication | 10*5 | 15 |
| / | Division | 10/5 | 2 |
| // | floor division | 10//3 | 3 |
| % | Modulus | 10%3 | 1 |
| ** | Exponent (power) | 2**3 | 8 |

→ Assignment operators: used to assign values to variables.

| Operator | meaning | Example | Equivalent |
|---|---|---|---|
| = | Assign | x=5 | x=5 |
| += | Add & assign | x+=3 | x=x+3 |
| -= | subt & assign | x-=3 | x=x-3 |
| *= | multi & assign | x*=3 | x=x*3 |
| /= | divide & assign | x/=3 | x=x/3 |
| //= | floor & assign | x//=3 | x=x//3 |
| %= | modulus & assign | x%=3 | x=x%3 |
| **= | power & assign | x**=3 | x=x**3 |

→ Comparision: used to compare values. Return true (or) false

| Operator | meaning | Example | Result |
|---|---|---|---|
| == | equal to | 5==5 | True |
| != | Not equal | 5!=3 | True |
| > | greater than | 5>3 | True |
| < | lesser than | 3<5 | True |
| >= | Greater (or) equal | 5>=5 | True |
| <= | lesser (or) equal | 3<=5 | false |

⇒ logical operators: Used for logical decisions
Works with conditions

| Operator | Meaning | Example | Result |
|---|---|---|---|
| and | True if both true | True & false | false |
| or | True if atleast one true | True (or) false | True |
| not | Reverse the condition | not True | false |

⇒ Membership Operators: used to test if a value exists in a sequence (list, string, tuple, set)

| Operator | meaning | example | Result |
|---|---|---|---|
| in | True if value present | 'a' in "apple" | True |
| not in | True if not present | 'b' not in "apple" | True |

Identity operators: used to compare memory location of two objects

| Operator | meaning | Example | Result |
|---|---|---|---|
| is | True if same object | x is y | True/false |
| is not | True if not same object | x is not y | True/false |

Ex:- x = [1,2,3]
y = x
z = [1,2,3]
print (x is y) # true (same object)
print (x is z) # false (same value, diff object)

③ Explain input and output operations in detail :

Input function & its default data type:

⇒ Input () function : Input () function are used to take input from the user

Syntax : Var = input ("Message")

It always returns data as string by default

⇒ Default data type: Data taken using input () is always stored as str (string type)

Type conversion (casting)

To convert input to other data types:

⇒ Convert to integer :

num = int (input ("Enter a number"))

⇒ Convert to float:

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

⇒ Convert to boolean (manually interpreted)

Ex : x = input ("Enter anything: ")

print (type(x)) # output: < class 'str' >

Type conversion while taking input:

⇒ input() always takes data as string (str) by default

→ To use numbers (for math), we must convert the input

→ Common conversions:

int () → converts input to integer

float () → converts input to decimal number

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

b = float (input ("Enter ~~ag~~ price : "))

⇒ without conversion:

x = input ("5") → "5" (string)

y = input ("3") → "3" (string)
x+y = "53" (string concatenation).

→ Taking multiple inputs:
Sometimes we need more than one input from the user
method 1:

a, b = input().split()
⇒ user enters values in one line
⇒ Default type is string
Example: a, b = input("Enter two numbers: ").split()
⇒ Type conversion with split():
If we need integer values:
a, b = map (int, input().split())

If we need float values:
x, y = map (float, input().split())
Ex: a, b = map (int, input("Enter two numbers: "))
print (a+b)
Here split() → breaks input by space
map() → converts each value

④ Discuss control statements and Decision-making statements in python
. Meaning and importance of control statements:
meaning: Control statements are special instructions in a programming language that control the flow of execution of a program. They decide which part of code runs, how many times it runs & under

Importance of control statements:

Decision Making: Control statements allow programme to make decisions and choose different actions based on conditions.

Repetition of tasks: loops help execute a block of code multiple times without writing it repeatedly

Better program flow & control: They allow development to control the order in which instructions are executed improving efficiency.

Reduce code length: Using loops and conditions reduce duplicate code, making the program shorter & cleaner.

Improves flexibility: programs becomes more flexible & adaptable to different inputs & situations.

Enhances program logic: They make it possible to implement real-life logic in software

Types:

=) Conditional (Decision-making statements):

used to check conditions & decide what to do

* if
* if-else
* elif

=) looping:

used to repeat a block of code multiple times

* for
* while

=) Jump statements:

used to change the normal flow inside loops

* break → stops the loop
* Continue → skips one iteration & continues

return → exists a function

8 Write an essay on python programming fundamentals

→ Role of programming in problem solving:

Programming plays an important role in solving problems because it helps us tell the computer what to do in a clear & step by step way:

→ Breaking problems into steps
→ Doing tasks automatically
→ Giving fast & correct results
→ Handling big amounts of information
→ Improving logical thinking

⇒ Python syntax Simplicity & readability:

Python is known for having a simple & clean syntax. This means the way we write python code

⇒ easy to understand
⇒ easy to write
⇒ close to normal english language
⇒ Simplicity in syntax
⇒ Readability : Readability means the code is easy to read & understand, even after many months

→ python code looks clean & neat, like plain English instructions.

Because of this:

⇒ develops can find errors easily
⇒ teams can work together better
⇒ programs are easier to maintain

Ex: if age > 18 :

Print ("Adult")

This reads like a simple english sentence.

Data types, operators, & input/output operations:

Data types: Data types define the type of data a variable can hold. They help the computer understand how to store and use the data.

Common data types in python:

=) int (integer)
=) float
=) decimal str (string)
=) bool (boolean)
=) list
=) Tuple
=) dict (dictionary)

Operators:

=) Arithmetic
=) comparision
=) logical
=) Assignment

① Program on movie ticket pricing:

```
age = int(input ("enter age:"))
is3D = int (input ("enter 1 if you are watching a
                    3D movie else enter 0:"))

if is3D == 1
    if age < 13:
        print ("ticket price = 200")
    elif age >= 13 and age <= 59 :
        print ("ticket price = 300")
    else:
        print (ticket price = 250")
```

```
if age < 13:
    print ("ticket price = 150")
elif age >= 13 and age <= 59
    print ("ticket price = 250")
else:
  print ("ticket price = 200")
```

② program on college attendence:

```
attendence = int (input ("enter attendence :- "))
medical_certificate = (input (" medical_certificate (Yes|No): "))
  iff attendence >= 75 or attendence >= 60 and
                                medical_certificate == "Yes":
        Print ("Allowed")
  else:
    print (" Not Allowed")
```

③ Progam on E-commerce Discount:

```
bilamo = int (input ("Enter bill amount : "))
isprime = int (input (" Enter 1 if you age prime
                                member else 0: "))

  if isprime == 1
     if bilamo >= 5000:
          dis = 0.25 * bilamo
     elif 2000 <= bilamo <= 4999:
          dis = 0.15 * bilamo
     else:
        dis = 0
     else:
        if bilamo >= 5000:
```

```
    dis = 0.20 * bilamo
elif 2000 <= bilamo <= 4999:
    dis = 0.10 * bilamo

else:
    dis = 0
amo = bilamo - dis
print ("final amount tobe paid is", amo)
```

④ Program on Smartphone Battery warning:

```
battery = int (input ("Enter battery: "))
is charging = int (input (" Enter 1 if phone is charging
                          else 0" ))

if ischarging == 0:
    if battery <= 20
        print ("Low Battery")

    elif 21 <= battery <= 80:
        print (" Normal Battery")
    else:
        print (" Full Battery")

elif ischarging == 1 :
        print ("charging")

else:
    print ("Enter 0 or 1")
```

⑤ Program on Driving license check

```
age = int (input ("Enter your age:"))
testparsed = int (input ("Enter 1 testparsed (1/0): "))
    if age >= 18 and test passed == 1 (or) age >= 60:
        print ("Eligible")
```

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

⑥ Program on Online food delivery:

```
amount = int(input("Enter amount:"))
isGold = (input("isGold (Yes/No): "))
distance = int(input("Enter distance : "))
if amount >=500 and isGold == "Yes":
    Print ("free Delivery")

elif distance > 10:
    Print ("Delivery charged")
```

⑦ Program on Bank Loan Approval:

```
salary = int (input("Enter salary:"))
credit_score = int( input ("Enter credit_score : "))
if salary >=30000 and credit_score >=700 or sal>50000:
    Print ("Loan Approved")

else:
    print ("Loan Rejected")
```

⑧ Program on Electricity Bill:

```
units = int(input("Enter number of units consumed: "))
    if units <=100:
        bill = units * 2
    elif units <=200:
        bill = 100 * 2 + (unit - 100) * 3

    else:
        bill = 100 * 2 + 100 * 3 + (units - 200) * 5
    Print (" final bill amount = ", bill)
```

⑧ Program on student Scholarship:

```
marks = int(input ("Enter marks: "))
~~family~~income = int (input ("Enter income:"))
single-parent = int (input ("Enter 1 if you have single
                Parent else 0: "))

if single-parent ==1 and marks >=85:
    print (" Eligible")
elif marks >=85 and income <50000
    print ("Eligible")

else:
    print ("not eligible")
```

10. Program on outline Exam Result:

```
theory = int(input ("Enter theory: "))
practical = int (input ("Enter practical: "))
total = theory + practical
if total >=100 or (theory >=40 and practical >=40):
    print ("pass")

else:
    print ("fail")
```

11. Program on Hotel Room pricing:

```
daystayed = int (input ("Enter number of days stayed:"))
isweekend = int (input ("Enter 1 if any day is a
                weekend else 0:"))
number = int (input ("Enter number of weekends in
                daystayed:"))

if is weekend == 0
    amount = daystayed * 3000
```

```
else:
    amount = number * 4000 + (daysstayed - number)* 3000
if daystayed > 3:
    discount = 0.15* amount
else:
    discount = 0
final amount = amount - discount
    Print ("Final bill : ", finalamount)
```

12. Program on Gaming level unlock.

```
Score = int (input ("Enter game score:"))
ispremium = int (input ("Enter 1 if you have premium
                        else 0: "))
used-cheat = int (input ("Enter 1 if you used-cheat
                        else 0: "))
if (score >= 100 (or) ispremium == 1) and used_cheat==0:
    print ("Next level unlocked")
elif used_cheat==1
    print ("Access denied")
else:
    print ("Next level is locked")
```

13. Program on Mobile Data Usage:

```
data used = int (input ("Enter data used: "))
unlimitedplan = int (input ("Enter 1 if you have unlimited plan
                        else 0"))
isRoaming = int (input ("Enter 1 if roaming is on else 0"))
if (dataused <= 2 or unlimpla == 1) and isroaming==0:
    Print ("Unlimited data")
```

```
else:
    print("unlimited data applies);
```

14 Program on office Entry system

```
idvalid = int(input("Enter 1 if id is valid else 0:"))
finpri = int(input("Enter 1 if fingerprint is valid else 0:"))
facescan = int(input("Enter 1 if facescan is valid else 0:"))
isholiday = int(input("Enter 1 if holiday else 0:"))
if idvalid == 1 and (finpri == 1 or facescan == 1):
    if isholiday == 1
        print("Accessdenied today is Holiday")
    else:
        Print("Enter into office")
    else:
        print("Acess denied")
```

15. Program on Movie Rating Display

```
average = float(input("Enter rating:"))
isedicho = int(input("Enter 1 if it is editor choice
                        else 0:"))
if isedicho == 1
        print("recommended")
elif average >= 8.5:
        print("Excellent")
elif average >=60 and average<=8.4:
        print("Good")
else:
        Print("Average")
```