

Handwritten

* Class Assessment-1 *

29/30

Page No.	
Date	

Q.1 Explain the difference between list & tuple in Python. Provide an example for each.

→ Lists : i) Lists are mutable - meaning you can modify their content.

ii) They are defined using square brackets '[]'.

iii) Lists are generally used for collections of items where the order & the ability to change the element are important for example L = ["shubham", 1, 2, 3, 4, 0.4, "Thorat"]

Tuples : i) Tuples are immutable - you cannot change their content.

ii) They are defined using parenthesis '()'.

iii) Tuples are suitable for fixed collections where the order of elements matters, but you don't want the data to be modified. For example T = (1, 2, 3, "Apple", "Banana")

Q.2 Describe the purpose of the set data type in python. Provide an example to illustrate its use.

→ Purpose : i) It is used to represent an unordered collection of unique elements.

ii) Primary purpose of a set is to perform mathematical set operation like union, intersection, difference & symmetric difference efficiently.

iii) Sets are mutable,

e.g. a = {1, 2, 3, 4, 5}

b = {3, 4, 5, 6, 7}

a.add(8)

b.remove(7)

Sets can be used for tasks such as combining, finding common elements, identifying differences & more.

Q.8. What is the key difference betw a float & an integer data type in python? Give an example where using a float would be more appropriate.

- • Integer ("int") : Represents whole numbers without decimal point. e.g. 2, 1, 5, 100.
- Float : Represent number with decimal points or numbers expressed in exponential form. e.g. 3.14, 0.5, 2e3

Distance = 10
Time = 2.5

Speed = distance / time
Point = (speed, distance, "m/s")

Q.4 How does dictionary data type in python differ from lists & tuples provide an example of dictionary & explain its structure.

- i) The dictionary data type in python differs from lists & tuples in that it is an unordered collection of key value pairs, where each key must be unique.
- ii) Dictionaries are defined must be using curly braces {}.
- iii) It useful for storing & retrieving data in a way that is easily accessible & efficient for example -

Personal info - { "Name": "Shubham", "Age": 21, "city": "Karad" }

Structure : i) Personal info has key such as name, age, city, the values associated with keys provide information about the person (e.g. Shubham, 21, Karad)
ii) keys & values are separated by Colons (':') with each key value pair.

Page No.	
Date	

Q.5. What is doc string & use of this string in Python.

- A docstring in python is a string literal used as a comment at the beginning of module, function, class or method definition. Its primary purpose is to provide documentation or information about the purpose & usage of the code.
- Use :- 1) Documentation 2) Accessibility 3) Interactive development

Q.6. Explain the purpose of the // operator in python. Provide example to illustrate its use.

- // :- Its purpose is the floor division operator. It performs division & returns the target integer that is less than or equal to the result. It discards the fractional part of the division result & produces an integer quotient.

e.g. $s = 10 // 3$
print(s)

In e.g 's' assigned the value of 3 because the floor division of 10 by 3 is 3 with remainder 1.

Q.7. Differentiate b/w the == & is operator in python provides examples demonstrate their usages.

$==$ Operator is equality operator
 is Operator is identity operator

i) The '==' operator checks if two values of the operands are equal. i) The 'is' operator checks if two operands refer to the same object in memory.

ii) It compares the content of the obj. ii) It tests for object identity, not just equality of content.

$a = [1, 2, 3]$

$b = [1, 2, 3]$

$result = a == b$

$print(result)$

$a == b$ returns 'True'

$a = [1, 2, 3]$

$y = [1, 2, 3]$

$result = a is y$

$print(result)$
 $a is y$ returns False.

Q.8. What is the use of the `+=` operator in python? Provide an example to demonstrate its functionality.

- The `+=` operator in python is used as an assignment operator & it is a shorthand for updating the value of variable by adding another value to it, it commonly used for in place addition & assignment.

Count = 25

Count += 2

Print (Count) Output = 7

`+=` Operator add the value '2' to variable Count

Q.9. Discuss the use of the `'in'` operator in python. Provide an example of how it can be used.

- i) The `'in'` operator in python is used to test whether a specified value is present in a sequence.

ii) It returns the Boolean Value ('True' & 'False') based on the existence of the specified element in the sequence.

e.g message = "Good morning Shubham"

& = "Shubham" in message

Print (s)

Output : True
because Shubham is present in string message.

Q.10. Explain the concept of the ternary operator (`x if condition else y`) in python. Provide example scenarios where it can be employed.

- i) The ternary operator in python, also known as the Conditional expression, provides a concise way to write a simple 'if-else' statement in a single line.

ii) Its syntax is `x if condition else y`. The expression

evaluates to 'x' if condition is true else otherwise evaluate to 'y'

e.g. `a=10
b=15
if a>b:
 print ("large number =a")
else:
 print ("large number = b")`

`Print ("large number =a")`

`else:`

`print ("large number = b")`

Q.11 What is the purpose of the if statement in python? Provide an example demonstrating the use of an if statement.

→ i) The 'if' statement in python is used for conditional execution code. It allows to specify a block of code to be executed only if a certain condition is true. If the condition is false, the code block is skipped or an alternative block may be executed if an 'else' clause is present. e.g. `Num = int(input ("Enter a number :"))
if number >0:`

`print ("The entered num is positive")
elif num <0:`

`print ("The entered num is negative")`

`else:`

`print ("The entered num is zero")`

Q.12 Describe the difference betn while & for loops in python. Give an example for each loop type.

→ i) While loop

ii) The 'while' loop in python is used to repeatedly execute a block to iterate over a sequence

of code as long as specified (such as list tuple or string) Condition is true. - of other iterable object.

ii) It keeps iterating until the Condition becomes false. ii) It executes the block of code for each element in the sequence

e.g.

Count = 1

while Count <= 5:

 print (Count)

 Count += 1

112

Color = ["red", "blue", "orange", "yellow"]

for Color in Colors:

Q.13 Explain the significance of the break statement in python. Provide a scenario where using break is appropriate.

→ The 'break' statement in python is used to exit loop prematurely before its normal termination condition is met, when executed 'break' immediately terminates the loop & the program continues with the next statement after to loop.

e.g. input = input ("Enter 'exit' to stop the loop: ")

if input. lower () == 'exit':

 print ("Exiting the loop")
 break

 print ("Continuing the loop")

Q.14 Discuss the role of the continue statement in python.

Provide a code snippet demonstrating its use.

→ i) the 'continue' statement in python is used to skip the rest of the code inside loop for the current iteration & proceed to the next iteration. It is particularly useful.

useful when you wants to bypass certain condition or code block without permanently terminating the entire loop
e.g.

For Num in range(1, 11):

If num % 2 == 0:

 print ("Skipping even num : (num)")

 Continue

 print ("processing odd num: (num)").

Q.15. How does the else clause in loop contribute to control flow in python? Provide an example illustrating the use of the else clause in a loop.

- i) The 'else' clause in a loop in python is executed when the loop condition becomes false. It is not executed if the loop is terminated by a 'break' statement.
- ii) The else clause in a loop provides a way to execute a block of code after the loop has completed its normal iteration.

e.g.

For num in range(2, 11):

 For i in range(2, int(num**0.5)+1):

 If Num % i == 0:

 print ("Num is Not a prime Number")

 break

 else :

 print ("Num is a prime number")