

*Answer Sheet  
961057102*

**INDIAN INSTITUTE OF TECHNOLOGY ROPAR**  
**GE 103: Introduction to Computing and Data Structures**  
**First Semester of Academic Year 2024-2025**  
**End Semester Examination**

Duration: 3 Hours [2:30 PM – 5:30 PM] Max Points: 50

Date: 6-MAY-2024

Instructions:

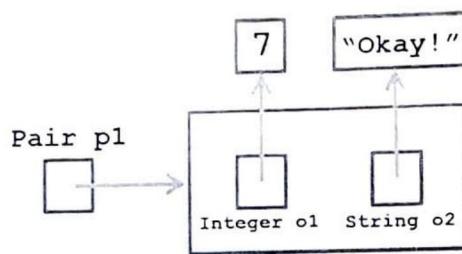
- There are 14 questions in the exam. All of the questions are mandatory.
- The points for each question are mentioned next to the question.
- Answer each question in the space provided for that question only.
- Structure your answer such that it does not go beyond the space allocated.
- No clarifications will be entertained during the examination. If you feel that a question is not clear, state your assumptions while answering.

Maximum Points	1	2	3	4	5	6	7	8
	3	3	4	4	5	2	4	4
Obtained Points								
Maximum Points	9	10	11	12	13	14	Total	
	3	3	4	4	3	4		
Obtained Points								

Please Turn Over.

Q1. The below diagram shows an object p1 of the Pair class containing two variables, one for each member variable associated with the object. Write the Object Oriented Python code corresponding to the diagram shown below. The following may be taken into account: [3 points]

- You need to declare a class called Pair, with two member variables, as shown below.
- Include a constructor function.



Ans.  $\text{Pair} = [7, \text{"Okay!"}]$

Def \_\_init\_\_(self, data) :

    data. self = data

    7. self = number

    okay!. self = string

Def \_\_type\_\_(self) :

    number = "7"

    string = "Okay!"

type1 = number ("The number is" 7).

type2 = string ("The string is Okay!")

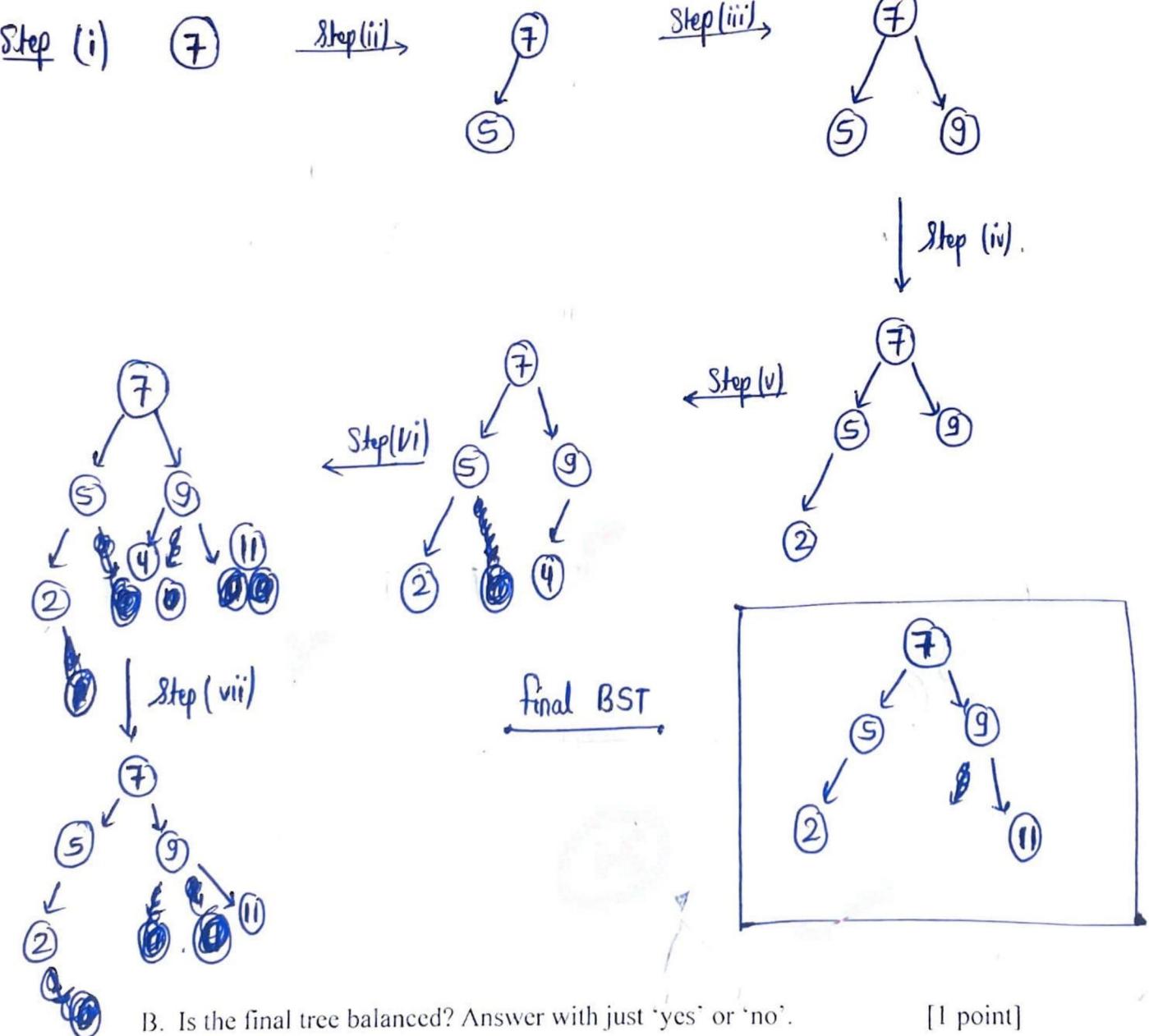
Print (type1)

Print (type2)

## Q2. Binary Search Trees

- A. Draw the resulting Binary Search Tree at each step after inserting and deleting the nodes in the following order - (i) insert 7, (ii) insert 5, (iii) insert 9, (iv) insert 2, (v) insert 4, (vi) insert 11, (vii) delete 4. Assume an empty tree at the beginning. Show the tree after performing each step. [2 points]

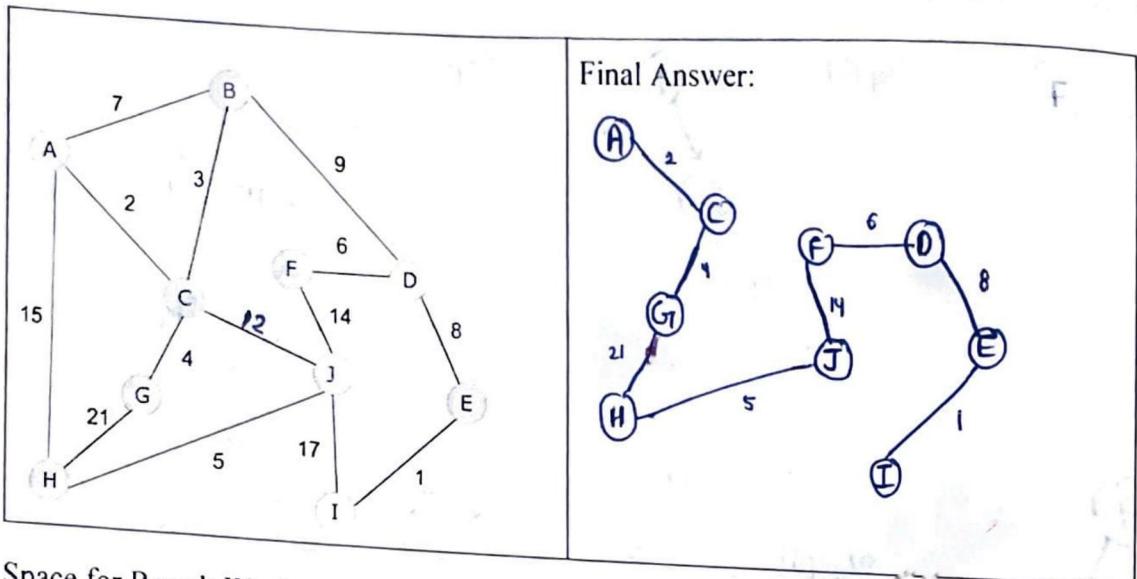
Binary search tree:



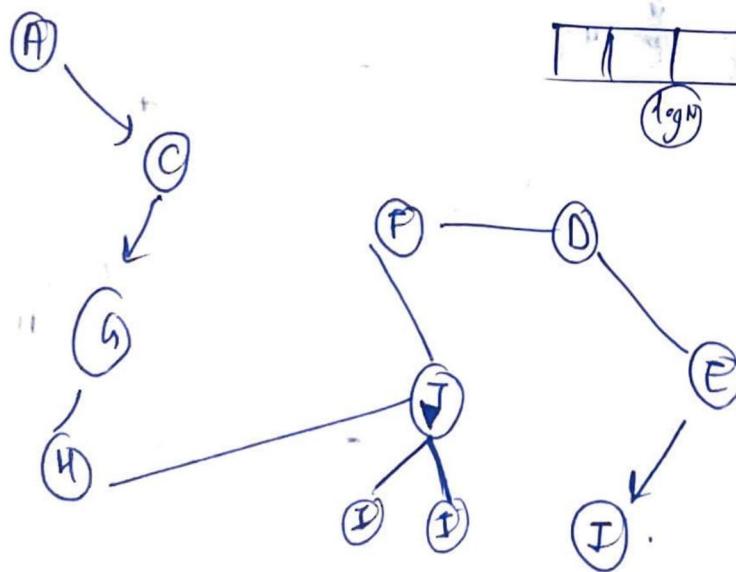
- B. Is the final tree balanced? Answer with just 'yes' or 'no'. [1 point]

- No

Q3. Draw the minimum spanning tree that is generated after applying Prim's algorithm to the graph below. Start with node A. Only draw the final minimum spanning tree. No explanation is required. [4 points]



Space for Rough Work:



Q4. Write down the expression of the worst case Big O complexity for the following operations. No explanation is necessary. [1 x 4 = 4 points]

A. Contains function: returns true or false depending on whether a particular object is a member of a list.

a. Sorted list implemented as an array. Answer:  $\log_2(N)$

b. Sorted list implemented as a single linked list (head reference only).

Answer: N

B. Insert function: add an object to the list in the appropriate place.

a. Unsorted list implemented as an array. Answer:  $O(n \log n)$

b. Sorted list implemented as an array. Answer:  $n \log n$

Q5. Circle the correct answer. Only one answer correct per question. [1 x 5 = 5 points]

a) Which of the following functions grows the fastest?

a. $n \log n$	(b) $2^n$
c. $\log n$	d. $n^2$

b) For a linked list implementation of a queue, if both front and rear have identical valid node values, the size of the queue is:

a. 0	(b) 1
c. 2	d. the answer cannot be determined

c) For the linked implementation of a stack, where are the push and pop operations performed?

a. Push in front of first element, pop the first element	(b) Push after last element, pop the last element
c. Push after last element, pop the first element	d. Push after first element, pop the first element

d) Which of the following does the binary heap implement?

(a) Binary search tree	b. Priority queue
c. Stack	d. None of the above

e) Which of the following would require the most extra space, on average?

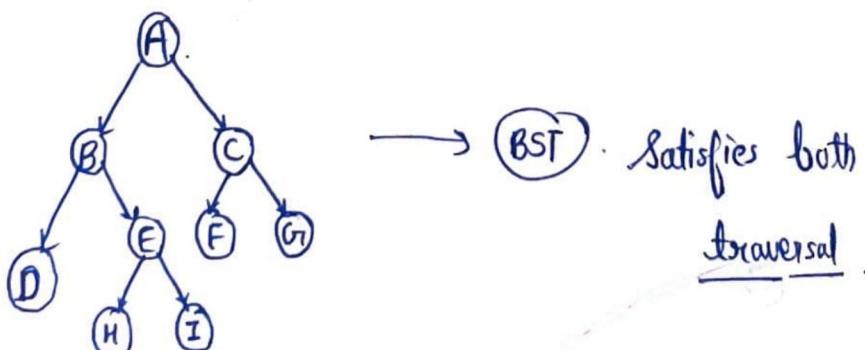
a. Bubble sort	(b) Merge sort
c. Quick sort	d. Selection sort

Q6. List two applications each for Stacks and Queues. [2 points]

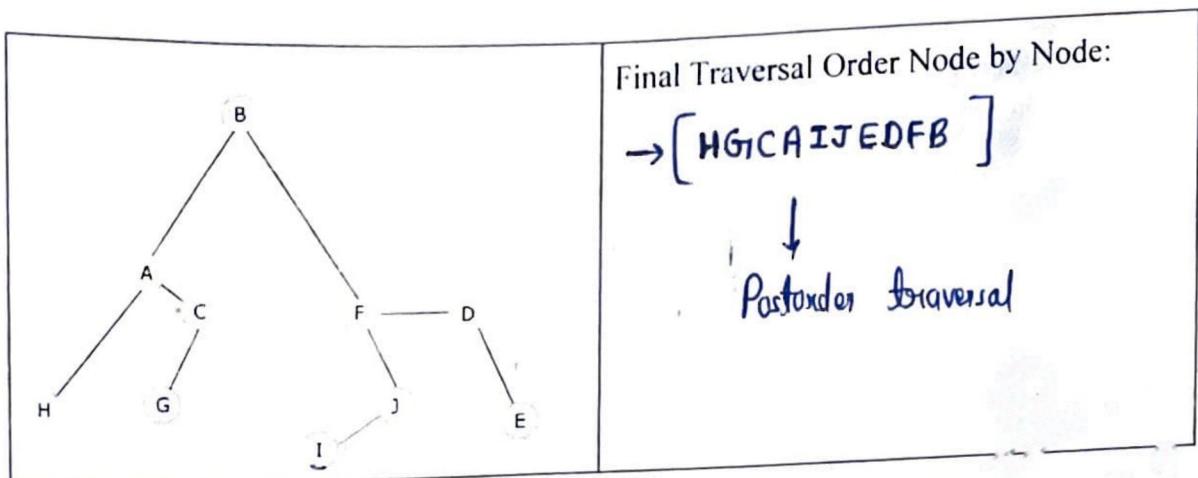
Stack (Application 1)	Stack (Application 2)
<del>LIFO and FIFO discipline is used</del>  ① Used to calls functions.	<del>Implemented</del>  ② Infix to postfix Conversion
Queue (Application 1)	Queue (Application 2)

Q7. Consider the results of the following binary tree traversals: (i) Inorder = DBHEIAFCG, and (ii) Preorder = ABDEHICFG.

Construct a single corresponding binary tree that satisfies both these traversal sequences. [4 points]



Q8. Consider the following graph. Draw the postorder traversal of the following binary tree, starting at node B. Show all the steps, and the contents of the stack at each step. [4 points]



Stepwise Answer:

Postorder : Left Subtree  $\rightarrow$  Right subtree  $\rightarrow$  Root

Step ① Visit the left subtree of B  $\rightarrow$  A then left subtree of A  $\rightarrow$  H  
 So, first node is (H)

Step ② Then Visit right subtree of A  $\rightarrow$  C then left subtree of C  $\rightarrow$  G  
 So, second node is (G).

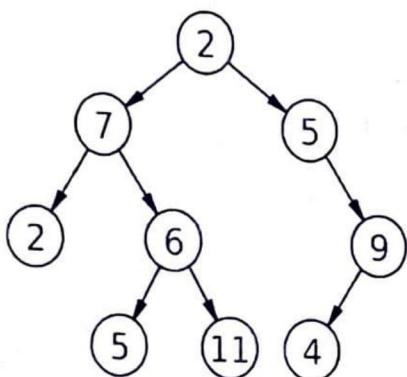
Step ③ Then follow reverse traversal So, (C) and (A) ~~are~~ are third and fourth node. Then Visit Right subtree of B  $\rightarrow$  F.

Step ④ Visit left subtree of J  $\rightarrow$  I. So (I) is the fifth node.  
 Then reverse traversal. So (J) is sixth node.

Step ⑤. Visit right subtree of F  $\rightarrow$  D, then Visit right subtree D  $\rightarrow$  E.  
 So (E) is the seventh node.

Step ⑥. Then follow reverse traversal. So, (D), (F) and (B) are the eighth, ninth and tenth node of Postorder traversal.

Q9. Given the definition of binary tree discussed in class, write the pseudocode for a recursive function named `SumLeaf( )`, that takes the root of a binary tree, and returns the sum of all the leaf nodes. For example, if you pass the root of the following binary tree, the function should return  $2 + 5 + 11 + 4 = 22$ . [3 points]



Definition: Binary tree are the trees which have almost two child nodes in structure.  
 $\therefore 0$  and  $1$  also can possible to be Binary tree.

Def `Sumleaf( )` :

If ~~pass~~  $2+5+11+4 == 22$  :

    return Pass.

else :

    return false.

If  $\text{Sum leaf} < 22$  or  $\text{Sum leaf} > 22$  :

    return Pass.

In this line we call `Sum leaf( )` which is called recursion.

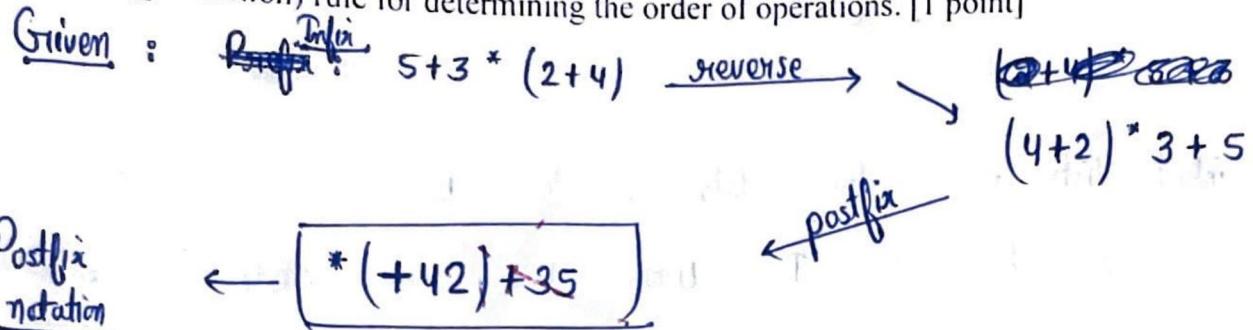
else :

    return false.

Print (`Sumleaf()`)

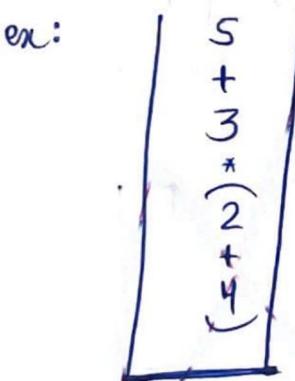
Q10. Postfix notation

- A. Write the following arithmetic expression in postfix notation:  $5 + 3 * (2 + 4)$   
 Use the BODMAS (Brackets, Order, Division, Multiplication, Addition, Subtraction) rule for determining the order of operations. [1 point]



- B. Next, use a stack to evaluate the above postfix expression. Show the push and pop operations used for each input character step by step, and the resulting stack after each step. [2 points]

By using Stack : Consider given Infix as stack

ex: 

Step ① We apply pop operations for each input character and make the stack empty.

Step ②. Then apply push operation for 5 and fill into stack.

Step ③ Similarly for 3, +, 2, 4, +, \* apply push operation and we obtain a postfix notation.

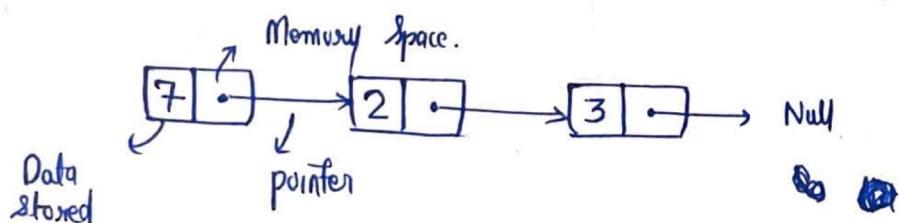
Q11. Linked Lists:

(a) What is a linked list data structure? Explain with the help of a pictorial representation. [2 points]

Ans. linked lists are linear data structure which is used to store memory and data. In which we can easily add or delete the memory and data.

- In Linked lists the insertion and deletion of memory is very easy rather than arrays,
- But they require a new memory space on insertion of each new data and Random access is not allowed.

Pictorial representation



(b) Give one advantage and one disadvantage of the linked list data structure over an array. [2 points]

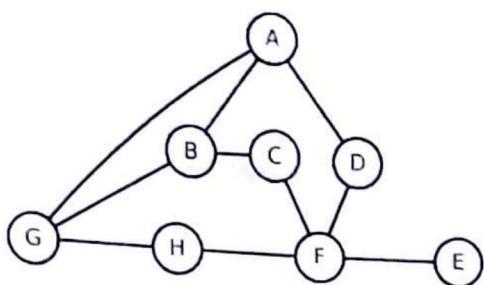
Advantage : Insertion and deletion of new data is very easy in linked lists.

But in array we have to displace all data to insert and delete new data which is difficult.

Disadvantage : In linked lists random access is not allowed

But in arrays it is random access is allowed.

Q12. Given the following graph:



Explain and show the steps for breadth-first search traversal using the ~~stack~~ data structure, starting from node A.

queue

[4 points]

Breadth - first Search traversal (BFS): ~~ADBGHFC~~ ADBGHFCE

Using queue data structure : In queue we use FIFO discipline. order.

Step ① Let's A be the first node in queue.

Step ② Then we visit all the nodes which is connected to A because they are in queue after A.

Step ③ Then we visit nodes After A → H → F → C → E.

follow queue.

Q13. Mutability:

(a) Give an example to illustrate the difference between mutable and immutable data types in Python. [2 points]

① Mutable : These data types are the data types in which we can able to do change in the initial value.

Ex: In lists.

$$L = [2, 3, 4]$$

So, We have to add 5 in given list, so give  
L.append(5)  
print(L) → Command to the Computer.

② Immutable : Ex: Dictionaries.

$$D = \{"a": "b"\}$$

We want to add "c" in dictionary but not possible.  
(b) How does the mutability of function arguments affect the behavior of functions in Python? [1 point]

Ans The mutability of function affect the behaviour of functions because we can able to change the value of that function according to our need so it affects the behavior of functions in python.

#### Q14. Access Modifiers:

- A. List and explain the different access modifiers available in Python with an example.  
[2 points]

Three types of access Modifiers : (i) Public (ii) private (iii) protected.

① Public : all Variables in python are public by default.

ex :  $x = 4$   
 $y = "ayyan"$

② private : They can't access directly but we can access them indirectly. ex : We use  $\underline{\underline{}}$ , double underscore.

③ protected : In this we able to access only the class generated by itself ex : use  $\underline{\underline{}}$  → single Underscore.

- B. What is inheritance property in the Python OOP paradigm? List two advantages of the inheritance property. [2 points]

Ans : The inheritance property in python oop Paradigm is that we can able to make new class from one class which is called Subclass of the given class. This is called + the property of inheritance in python.

advantages : ① This helps us to solve any problem in easy way.  
② We can make ~~two~~ Subclass from one class.