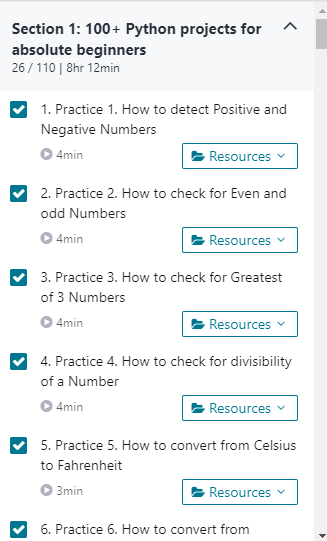
**DAILY ONLINE ACTIVITIES SUMMARY**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Date:** | **21/06/2020** | | | | **Name:** | **Imran Khan** | |
| **Sem & Sec** | **8th A** | | | | **USN:** | **4AL16CS040** | |
| **Online Test Summary** | | | | | | | |
| **Subject** | | **---** | | | | | |
| **Max. Marks** | | **--** | | **Score** | | **----** | |
| **Certification Course Summary** | | | | | | | |
| **Course** | **python** | | | | | | |
| **Certificate Provider** | | | **udemy** | **Duration** | | | **8 HOURS** |
| **Coding Challenges** | | | | | | | |
| Problem Statement:   C Program on BST. | | | | | | | |
| **Status: Solved** | | | | | | | |
| **Uploaded the report in Github** | | | | **yes** | | | |
| **If yes Repository name** | | | | **Imran040** | | | |
| **Uploaded the report in slack** | | | | **yes** | | | |

**Certification Course Details**:



**Coding Challenges Details**:

**program1:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
|  | |
| INT\_MAX = 4294967296 | |
|  | | INT\_MIN = -4294967296 |
|  | |  |
|  | | class Node: |
|  | |  |
|  | | # Constructor to create a new node |
|  | | def \_\_init\_\_(self, data): |
|  | | self.data = data |
|  | | self.left = None |
|  | | self.right = None |
|  | |  |
|  | |  |
|  | | # Returns true if the given tree is a binary search tree |
|  | | # (efficient version) |
|  | | def isBST(node): |
|  | | return (isBSTUtil(node, INT\_MIN, INT\_MAX)) |
|  | |  |
|  | | # Retusn true if the given tree is a BST and its values |
|  | | # >= min and <= max |
|  | | def isBSTUtil(node, mini, maxi): |
|  | |  |
|  | | # An empty tree is BST |
|  | | if node is None: |
|  | | return True |
|  | |  |
|  | | # False if this node violates min/max constraint |
|  | | if node.data < mini or node.data > maxi: |
|  | | return False |
|  | |  |
|  | | # Otherwise check the subtrees recursively |
|  | | # tightening the min or max constraint |
|  | | return (isBSTUtil(node.left, mini, node.data -1) and |
|  | | isBSTUtil(node.right, node.data+1, maxi)) |
|  | |  |
|  | | # Driver program to test above function |
|  | | root = Node(4) |
|  | | root.left = Node(2) |
|  | | root.right = Node(5) |
|  | | root.left.left = Node(1) |
|  | | root.left.right = Node(3) |
|  | |  |
|  | | if (isBST(root)): |
|  | | print ("Is BST") |
|  | | else: |
|  | | print ("Not a BST") |