|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **10** | | | | | | | |
| **5** | | | | **20** | | | |
| **3** | | **7** | |  | | **25** | |
|  |  |  | **8** |  |  |  |  |

1 private void preorderTraverse(TreeNode current) {

2 if (current == null) {

3 return;

4 }

5 System.out.print(current.getData() + ", ");

6 preorderTraverse(current.getLeft());

7 preorderTraverse(current.getRight());

8 }

**OUTPUT: 10, 5, 3, 7, 8, 20, 25,**

**TRACE WHEN CURRENT = ROOT**

1 1 Go to 2.

2 2 False. Go to 5.

3 5 Print “10, “. Go to 6.

4 6 Traverse the 5 node. Go to 1.

5 1 Go to 2.

6 2 False. Go to 5.

7 5 Print “5, “. Go to 6.

8 6 Traverse the 3 node. Go to 1.

9 1 Go to 2.

10 2 False. Go to 5.

11 5 Print “3, “. Go to 6.

12 6 Traverse null. Go to 1.

13 1 Go to 2.

14 2 True. Go to 3.

15 3 Return to after 12.6.

16 7 Traverse null. Go to 1.

17 1 Go to 2.

18 2 True. Go to 3.

19 3 Return to after 8.6.

20 7 Traverse the 7 node. Go to 1.

21 1 Go to 2.

22 2 False. Go to 5.

23 5 Print “7, “. Go to 6.

24 6 Traverse null. Go to 1.

25 1 Go to 2.

26 2 True. Go to 3.

27 3 Return to after 24.6.

28 7 Traverse the 8 node. Go to 1.

29 1 Go to 2.

30 2 False. Go to 5.

31 5 Print “8, “. Go to 6.

32 6 Traverse null. Go to 1.

33 1 Go to 2.

34 2 True. Go to 3.

35 3 Return to after 32.6.

36 7 Traverse null. Go to 1.

37 1 Go to 2.

38 2 True. Go to 3.

39 3 Return to after 4.6.

40 7 Traverse the 20 node. Go to 1.

41 1 Go to 2.

42 2 False. Go to 5.

43 5 Print “20, “. Go to 6.

44 6 Traverse null. Go to 1.

45 1 Go to 2.

46 2 True. Go to 3.

47 3 Return to after 44.6.

48 7 Traverse the 25 node. Go to 1.

49 1 Go to 2.

50 2 False. Go to 5.

51 5 Print “25, “. Go to 6.

52 6 Traverse null. Go to 1.

53 1 Go to 2.

54 2 True. Go to 3.

55 3 Return to after 52.6.

56 7 Traverse null. Go to 1.

57 1 Go to 2.

58 2 True. Go to 3.

59 3 Return to after 40.7.

60 8 Done.