Day 15 - 7/5/2025

**Task 1**: Create a node  for a tree and include a constructor.

class Task001 {

   int value;

   Task001 left, right;

   Task001(int item) {

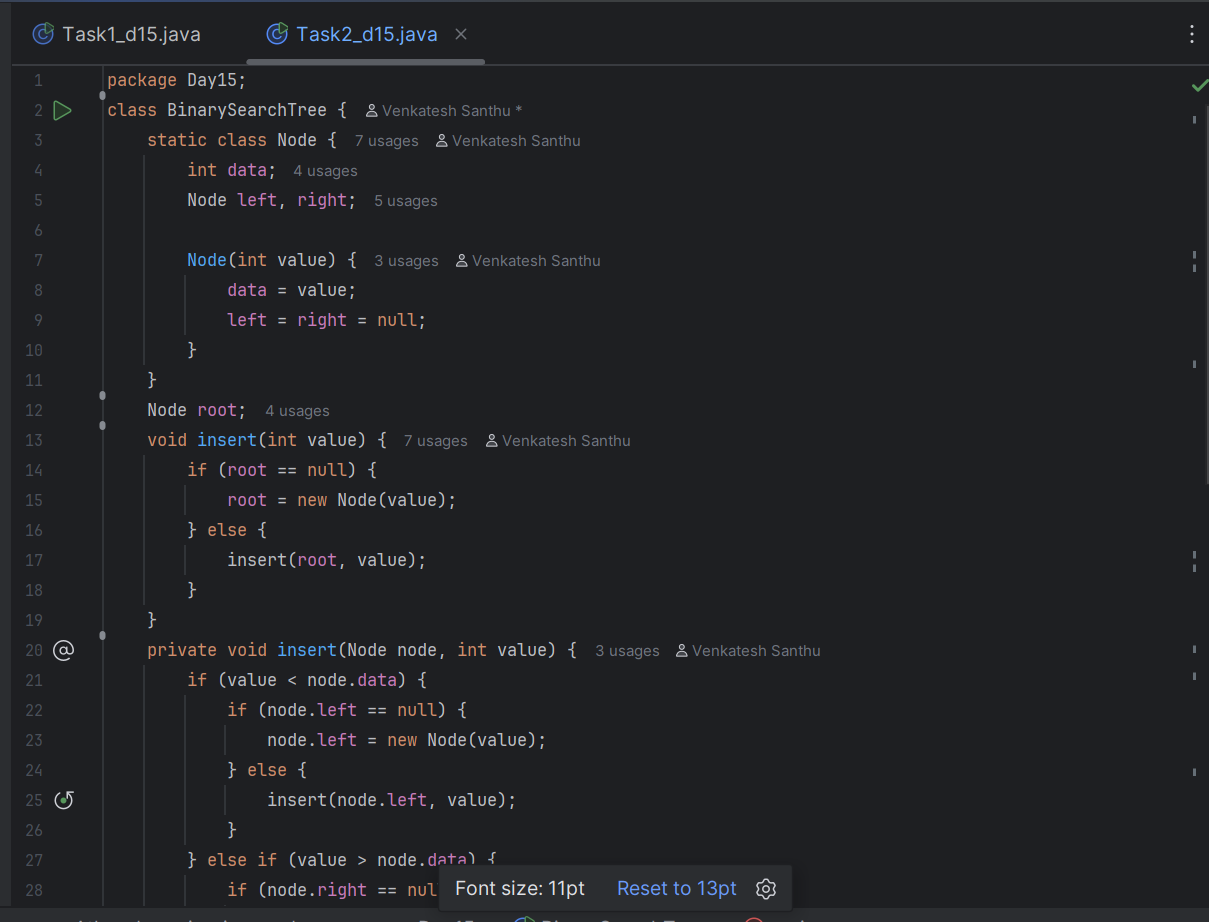
       value = item;

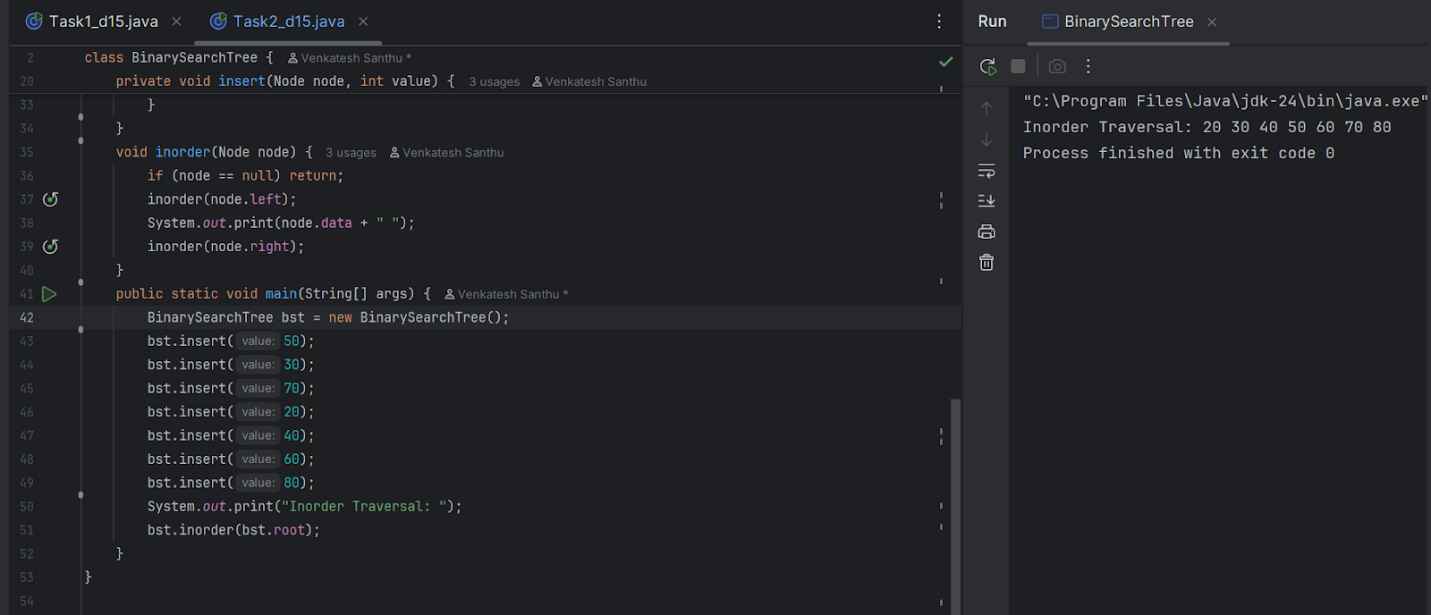
       left = right = null;

   }

}

**Task 2:** Create a class named Binary Search tree in which you have 2 insert operations





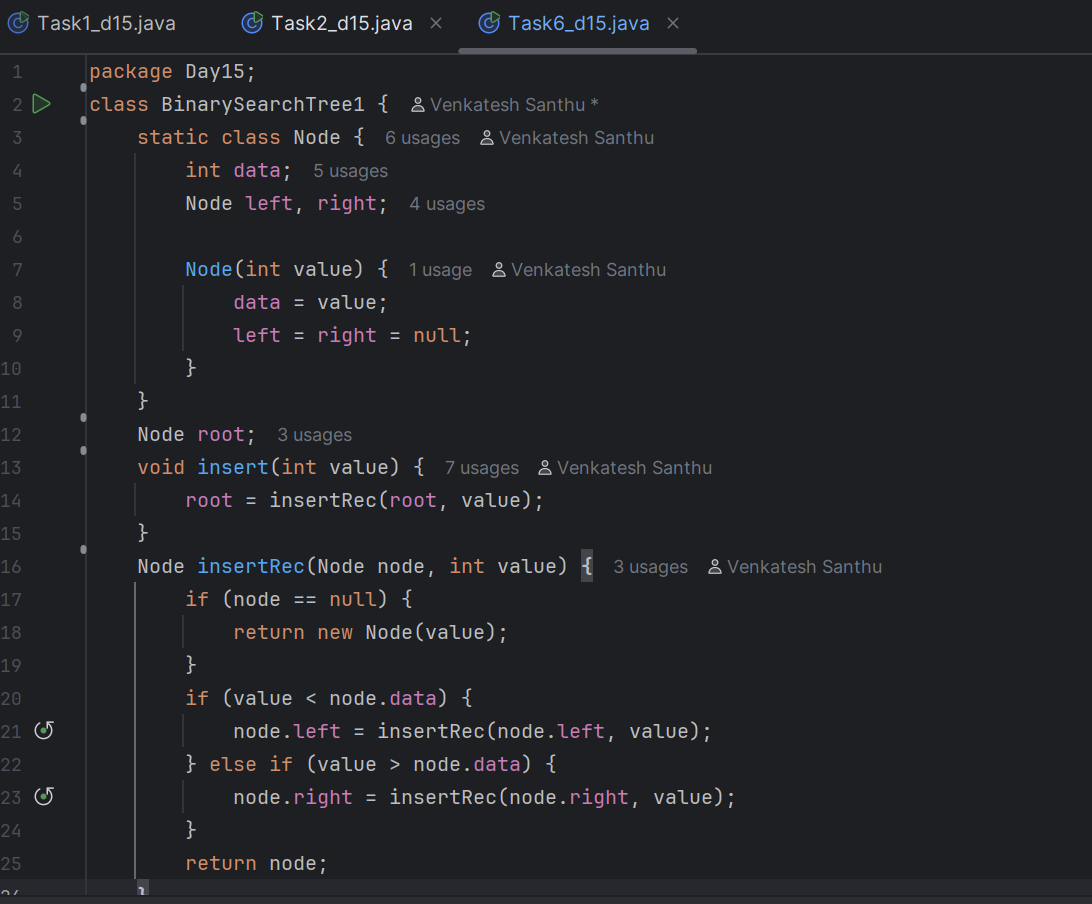
**Task 5**: Applications of Trees

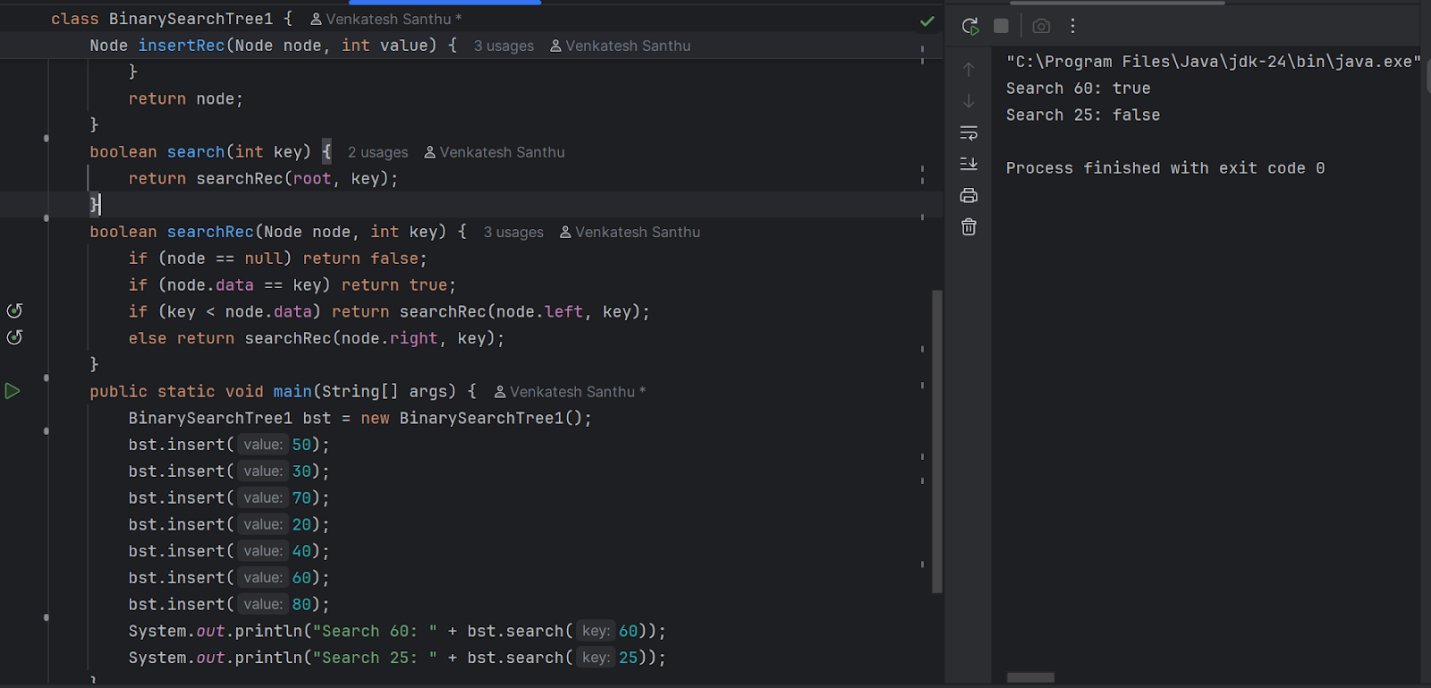
Trees are non-linear data structures. They are particularly useful for representing hierarchical data, efficient searching and sorting, and implementing various algorithms. Their applications are:

1. File Systems
2. Organizational charts
3. Decision trees
4. Data base indexing
5. Syntax trees
6. Data bases

**Task 6**: Create  a binary search operation on tree

Hint: Create a node





**Task 7**: Types of binary trees:

1. Rooted binary tree: It has root node & each node has at most 2 children.
2. Full/Strictly binary tree: A binary tree in which every node has either 0 or 2 children
3. Complete/Perfect Binary tree: every internal node has 2 children & all leaf nodes are at the same level.
4. Almost complete binary tree: All the levels are completely filled except the last level
5. Skewed binary tree: All the nodes has only one child except one node

**Task 8:** Applications of Graphs

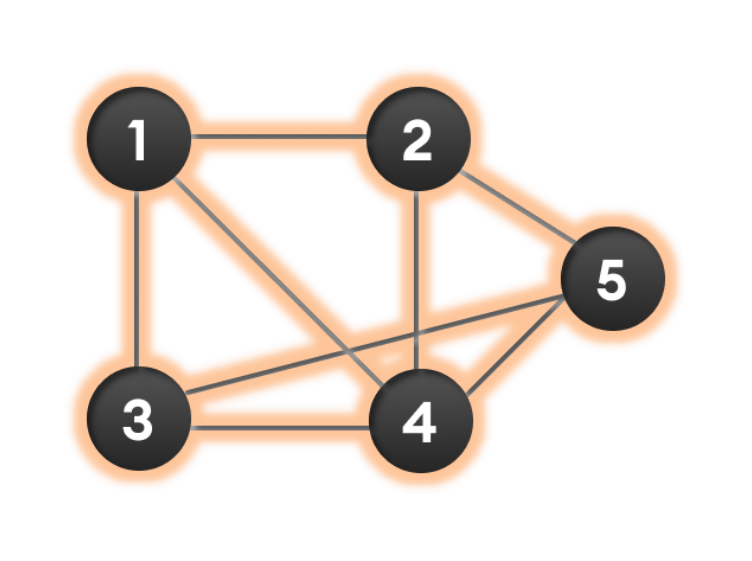
Graphs are non linear data structures. They has various applications,

1. Navigation (maps) system
2. Social networks
3. Search engines
4. Game developments

**Task 9:** Types of Graphs

1. Undirected graph: The graphs edges are not directing
2. Directed graph: The graph whose edges are directing
3. Complete graph: Graph whose V node is adjacent to all other nodes present
4. Regular Graph: Graph whose nodes are adjacent to each other
5. Cycle Graph: A graph which is closed and forms a circular shape. Its first and last nodes are the same.
6. Acrylic graph: A graph which does not have cycle is called acrylic graph
7. Weighted graph: Graph whose edges are assigned a non negative value.

Task10:



Wap to display a graph edges .in the below order no od edges 8 and no of vertex 5

1 - 2

1 - 3

1 - 4

2 - 4

2 - 5

3 - 4

3 - 5

4 - 5

class Graph {

   class Edge {

       int src;

       int dest;

       Edge(int s, int d) {

           src = s;

           dest = d;

       }

   }

   int vertex;

   int edge;

   Edge[] edges;

   Graph(int v, int e) {

       vertex = v;

       edge = e;

       edges = new Edge[e]; // Array to store edges

   }

   void displayEdges() {

       for (int i = 0; i < edge; i++) {

           System.*out*.println(edges[i].src + " - " + edges[i].dest);

       }

   }

}

public class Task010 {

   public static void main(String[] args) {

       int v = 5;

       int e = 8;

       Graph g = new Graph(v, e);

       g.edges[0] = g.new Edge(1, 2);

       g.edges[1] = g.new Edge(1, 3);

       g.edges[2] = g.new Edge(1, 4);

       g.edges[3] = g.new Edge(2, 4);

       g.edges[4] = g.new Edge(2, 5);

       g.edges[5] = g.new Edge(3, 4);

       g.edges[6] = g.new Edge(3, 5);

       g.edges[7] = g.new Edge(4, 5);

       g.displayEdges();

   }

}

