AVL Tree Implementation

Assignment 3 Question 2 implements a self-balancing AVL tree in C++. The implementation includes insertion, deletion, and various balancing operations to maintain the AVL tree properties.

Project Structure

The project consists of two main files:

`avl.hpp`
- Header file containing the AVL node structure definition
- Defines the basic structure for tree nodes with:
  - Element (integer value)
  - Left and right child pointers
  - Height information

`avl.cpp`
Main implementation file containing all the AVL tree operations:

1. **Core Operations**
   - `insert`: Adds a new value to the tree
   - `remove`: Deletes a value from the tree
   - `print`: Displays the tree structure

2. **Helper Functions**
   - `height`: Gets node height
   - `updateHeight`: Updates height after modifications
   - `getBalance`: Calculates balance factor
   - `findMin`: Finds minimum value in a subtree

3. **Balancing Operations**
   - `rotateLeft`: Left rotation for balancing
   - `rotateRight`: Right rotation for balancing
   - `balance`: Main balancing function

Compilation and Usage

Compiling the Code
```bash
g++ avl.cpp
```

Running the Program
```bash
./a.out input1.txt
```

The program reads commands from an input file. Supported commands:
- `insert <value>`: Inserts a number into the tree
- `delete <value>`: Removes a number from the tree
- `print`: Displays the current tree structure

Input File Format Example
```
insert 100
insert 30
print
delete 30
```

print
```

### Output Format
The tree is printed in a hierarchical format:
```
30
|l_20
|r_100
  |l_50
```
Where:
- Each line shows a node's value
- `|l_` indicates a left child
- `|r_` indicates a right child
- Vertical bars (`|`) show the depth level

## Implementation Details

### Balancing
The implementation maintains the AVL property by:
1. Tracking height information for each node
2. Calculating balance factors during modifications
3. Performing rotations when the balance factor exceeds ±1
4. Handling all four imbalance cases:
   - Left-Left: Single right rotation
   - Left-Right: Left-right double rotation
   - Right-Right: Single left rotation
   - Right-Left: Right-left double rotation

## Error Handling
- The program validates input file format
- Duplicate values are ignored during insertion
- Invalid commands are reported as errors

## Notes
- The implementation assumes integer values
- Height information is automatically maintained
- Tree is printed after each print command
- Memory is properly managed to prevent leaks
```