Assignment III: AVL Tree Implementation

Student Name: ____________________________

Student ID: ____________________________

Instructions:

You are required to implement an AVL Tree to store double values using Object-Oriented Programming (OOP) principles in C++.
Your program should:

- Define a class for **AVLTree** that supports insertion, deletion, and searching of double values.
- Implement balancing operations to maintain the AVL property after insertions and deletions.
- Study and discuss the effect of floating-point precision on the correctness and balancing of the AVL tree (e.g., how comparing double values may affect insertions, deletions, and tree structure). Provide examples or test cases illustrating potential issues and how to handle them.
- Implement traversal of the tree in-order, pre-order, and post-order.
- Implement a method to merge two AVL trees without losing the AVL property:
  1. Convert both trees to sorted arrays using in-order traversal.
  2. Merge the two sorted arrays into one efficiently.
  3. Construct a new AVL tree from the merged sorted array.
  4. Include several test cases to demonstrate the merging process and the correctness of the resulting AVL tree.
- Demonstrate encapsulation and other OOP concepts such as method overloading and operator overloading where applicable.

Submission Requirements:

- Include your well-commented source code in the report.
- Provide sample test cases and their outputs.
- Briefly explain your design choices and how OOP principles are applied.
- Ensure your code is well-structured and follows best practices.

A Introduction

This document illustrates how to use the provided LaTeX class and environments for your programming assignment submissions. Use the `codelisting` environment for your code, and the `testcase` environment for worked examples.

B Sample Code Listing

Below is an example of how to include C++ code in your report:

```
// Example: Hello World in C++
#include <iostream>
using namespace std;
```

```
int main() {
    cout << "Hello,␣world!" << endl;
    return 0;
}
```

C Test Cases

Here are some sample test cases you can include in your report:

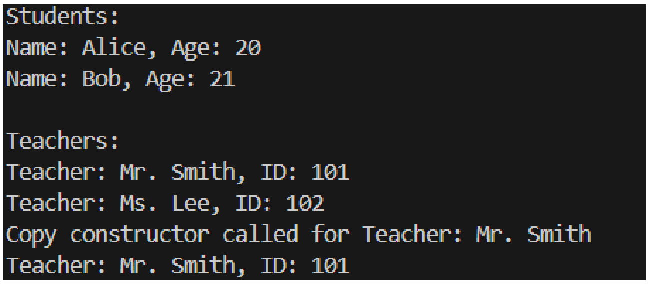
Test Case: Swapping Two Numbers in C++

```
int a = 5, b = 10;
swap(a, b);
cout << "a␣=␣" << a << ",␣b␣=␣" << b << endl;
\end{codelisting   }

Output:
\begin{codelisting}
a = 10, b = 5
```

D Inserting Figures

You can include figures in your report using the `figure` environment. Here is an example:



```
Students:
Name: Alice, Age: 20
Name: Bob, Age: 21

Teachers:
Teacher: Mr. Smith, ID: 101
Teacher: Ms. Lee, ID: 102
Copy constructor called for Teacher: Mr. Smith
Teacher: Mr. Smith, ID: 101
```

Figure 1: Sample Image