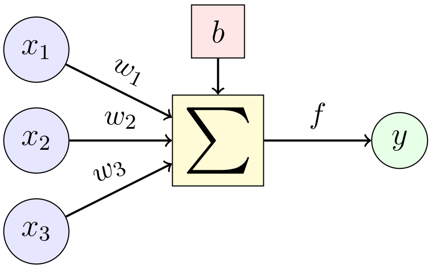
Final Project

Instructions:

Objective:
Implement a C++ class to represent a perceptron, a fundamental building block of neural networks. You are **not** required to implement a training algorithm.

1. **Perceptron Class:**
   - Implement a class that models a single perceptron.
   - The perceptron should support at least two classical activation functions (e.g., step, sigmoid, ReLU, tanh). More activation functions are encouraged.
   - The perceptron should accept a vector of inputs and corresponding weights, plus a bias term.
   - The class should have methods to compute the output given inputs and weights, and to set or get the weights and bias.
   - Include a constructor to initialize the perceptron with weights and bias.

2. **What is a Perceptron?**
   A perceptron is a simple computational unit that takes several inputs, applies weights to them, sums the result, adds a bias, and passes the sum through an activation function to produce an output. It is the basic unit of a neural network.

   In the diagram above, x_i are the inputs, w_i are the weights, b is the bias, and f is the activation function applied to the weighted sum of inputs plus bias to produce the output y of the perceptron:

   $$y = f\left(\sum_{i=1}^{n} w_i x_i + b\right)$$

3. **Network of Perceptrons:**
   - Design a data structure to represent a layer of perceptrons, and a network composed of multiple layers (as in a feedforward neural network).
   - *Hint:* You may use `std::vector` to store perceptrons in a layer, and a vector of layers to represent the network.

4. **Report:**
   - Document your design choices, especially the data structures used.
   - Include the C++ code for your main classes.
   - Provide simple validation or test cases to demonstrate your implementation.

A 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;
}
```

B Test Cases

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

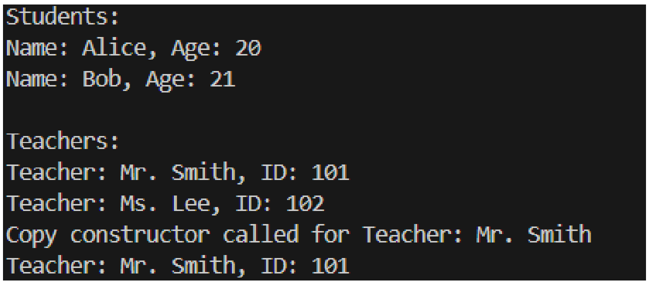
Test Case: Swapping Two Numbers in C++

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

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

C 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