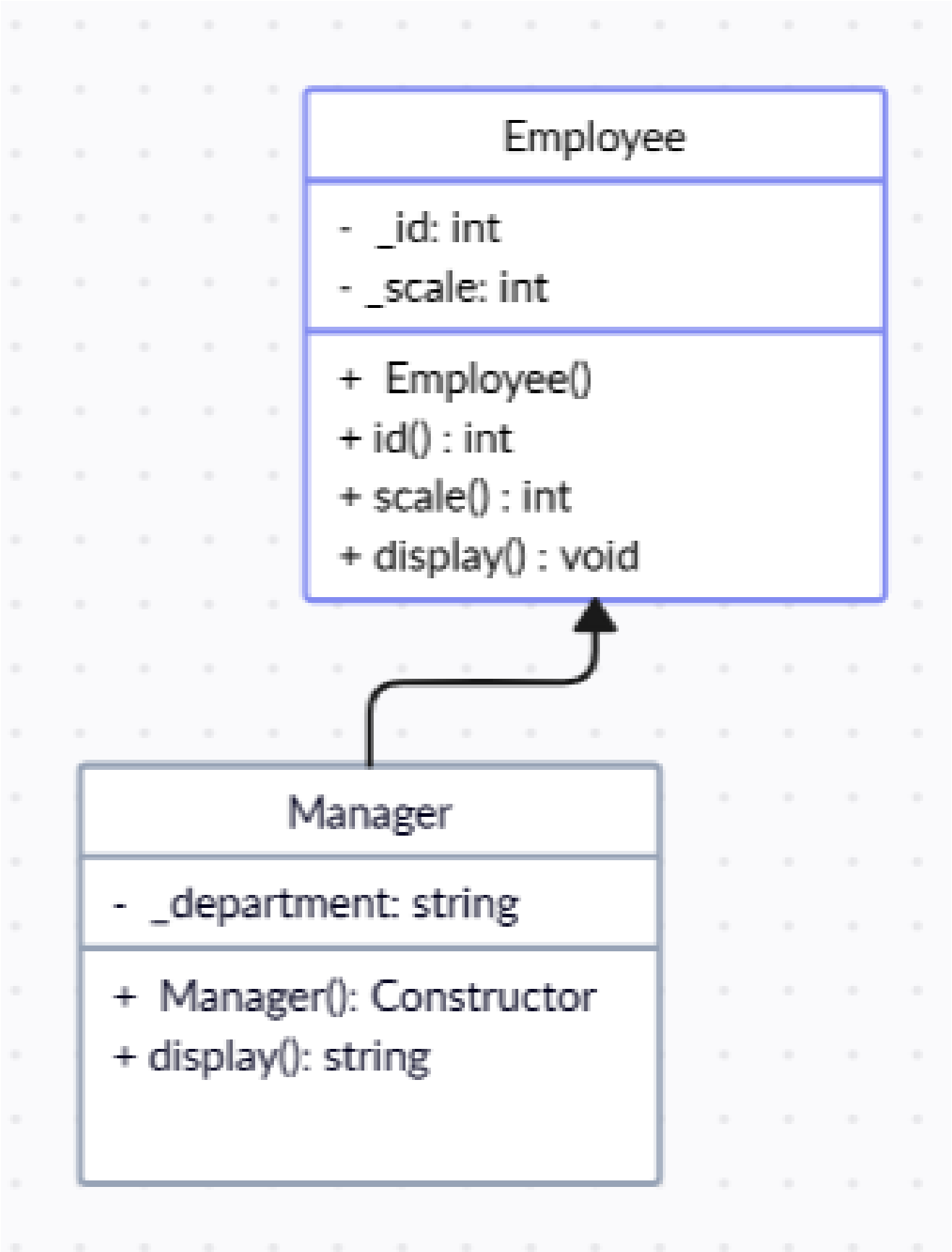
1. Write a class Employee that contains attributes of employee id and his scale. The class contains member functions to input and show the attribute. Write a child class Manager that inherits Employee class. The child id has attributes of manager id and his department. It also contains the declaration of member functions to input and show its attributes. Also, draw the UML class diagram of the code.

Solution

```cpp
#include <iostream>
using namespace std;

#define ll long long
#define clli const ll int
#define cs const string

class Employee{
    private:
        ll int _id, _scale;

    public:
        // Constructor
        Employee(ll int id, ll int scale) : _id(id), _scale(scale){

        }

        // Getters
        ll int id(){return _id;}
        ll int scale(){return _scale;}

        // display
        void display(){
            printf("Employee(id = %lli, scale = %lli)\n", _id, _scale);
        }
};

class Manager : public Employee{
    private:
        string _department;
    public:
        // Constructor
        Manager(clli& id, clli& scale, cs& department) : Employee(id, scale), _department(department){}

        // display
        void display(){
            printf("Manager(id = %lli, scale = %lli, department = %s)\n", id(), scale(), _department);
        }
};
```


Employee
- _id: int
- _scale: int
+ Employee()
+ id() : int
+ scale() : int
+ display() : void
Manager
- _department: string
+ Manager(): Constructor
+ display(): string

2. Create the UML class diagram of the following C++ code.

Solution