# POO (11:00-12:00)

Write the class **Function** so that the following code

```
#include "Office.h"

int main()
{
    Office asociatie;

    Owner* new_owner = new Owner(1200);
    asociatie.add(new_owner);

    Renter* new_renter = new Renter(300);
    asociatie.add(new_renter);
    asociatie.add((Neighbour*)new Owner(500));
    asociatie.add(new Renter(1300));

    try {
        asociatie.add(new_owner);
    }
    catch (const std::exception& e) {
        std::cout << e.what() << std::endl;
    }

    std::cout << "Total rent to collect: " << asociatie.collect_rent() << std::endl;

    try {
        std::cout << "Property tax to collect from owner: " <<
Office::collect_property_tax(new_owner) << std::endl;
        std::cout << "Property tax to collect from owner: " <<
Office::collect_property_tax(new_renter) << std::endl;
    }
    catch (const std::exception& e) {
        std::cout << e.what() << std::endl;
    }

    auto [renters, owners] = asociatie.get_statistics();
    std::cout << "Office has " << renters << " renters and " << owners << " owners.";
}
```

compiles and upon execution prints the following to the screen:

```
Neighbour already in list!
Total rent to collect: 1600
Property tax to collect from owner: 1200
Property tax to collect from owner: Neighbour is not an owner!
Office has 2 renters and 2 owners.
```

Carefully read the main function to deduce what methods/operators should be included in Office class.

**Observations:**

- Use STL to solve this problem
- There should be only one implementation for **add** function
- Use C++17 for compiling and testing (Project - <project_name> Properties - General - C++ Language Standard)

**Grading (informative):**

| | | |
|---|---|---|
| **G1** | Organize your project in 3 files: **main.cpp**, **Office.h** and **Office.cpp** | 2p |
| **G2** | Correct implementation of the required classes | 3p |
| **G3** | Destructor to clear memory | 2p |
| **G4** | Method add | 4p |
| **G5** | Validate add using exceptions | 3p |
| **G6** | Method collect_rent | 2p |
| **G7** | Method collect_property_tax | 2p |
| **G8** | Validate collect_property_tax using exceptions | 3p |
| **G9** | Method get_statistics | 6p |
| **G10** | Code compiles and upon execution prints the correct output | 3p |