Mechatronic Engineering

Object Oriented Programming and Software Engineering
Laboratory instruction 8
C++ introduction

AGH Kraków, 2021

Materials created for educational purposes.
Dedicated for students attending Software Engineering course.
Author would appreciate any feedback regarding errors of any kind found in the instruction script.
Please report those to the following email address: danielt@agh.edu.pl

Table of contents

1. Friend function.

A friend function is a function that has access to all members of a class (also private and protected) which has declared friendship with it. To create a friendship between a function and a class, add the appropriate annotation in the class syntax:

```
friend data_type name_of_function (class_name & element_name);
```

To refer to a class member, use the following:

```
object.element
```

Example:

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

class door; //declaration of the class
class home {

private:
   bool flowerHydr;

public:
   home (bool v) { flowerHydr = v;}

  void dayPassed() {
  flowerHydr = 0;
  cout << "day has passed, flower needs water"<<endl;
  }

  void ownerHydFlower() {
  flowerHydr = 1;
  cout << "flower has been hydrated by the house owner"<<endl;
  }

   friend void gardener(home& h, door& d);
};

class door {

bool lock;
```

```
public:

door(bool v) {lock = v;}


   void openDoor() {
   lock = 0;
   cout << "door has been opened" << endl;
   }

   void closeDoor() {
   lock = 1;
   cout << "door has been closed" << endl;
   }

   friend void gardener(home& h, door& d);
};

void gardener(home & h, door & d) {
    d.openDoor();
    h.flowerHydr = 1;
    cout << "flower has been hydratet by gardener"<<endl;
    d.closeDoor();
}


main() {
    home h1(0);
    door d1(1);

    h1.ownerHydFlower();
    cout << "Owner: It is time for adventure! Gardener
    will take care of my flower!" << endl;
    d1.openDoor();
    cout << "Owner left" << endl;
    d1.closeDoor();
    h1.dayPassed();
    gardener(h1, d1);


return 0;
}
```

---

2. Friend classes.

The idea of friend classes is and extension of friend functions, i.e. a friend class has a full access to private members of a class in which the friendship is declared. The friend class declaration is done in the class that will provide its private members.

To connect a class with other class to create a friend class (it has to be implemented inside the class that want to have a friend):

```
friend class name_of_class;
```

Example:

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

class door; //declaration of the class
class gardener;
class home {

private:
    bool flowerHydr;

public:
    home (bool v) { flowerHydr = v;}

    void dayPassed() {
   flowerHydr = 0; cout << "day has passed, flower needs water"<<endl;
    }

    void ownerHydFlower() {
   flowerHydr = 1; cout << "flower has been hydrated by the house
   owner"<<endl;
    }

    friend class gardener;
};

class door {
```

```
bool lock;

public:

   door(bool v) {lock = v;}

   void openDoor() {
   lock = 0;
   cout << "door has been opened" << endl;
    }

   void closeDoor() {
   lock = 1;
   cout << "door has been closed" << endl;
    }
    friend class gardener;
};

class gardener {
public:
   void gardening(home & h, door & d) {
   d.openDoor();
   h.flowerHydr = 1;
   cout << "flower has been hydratet by gardener"<<endl;
   d.closeDoor();
    }
};

main() {
    home h1(0);
    door d1(1);
    gardener g1;

    h1.ownerHydFlower();
    cout << "Owner: It is time for adventure! Gardener
    will take care of my flower!" << endl;
    d1.openDoor();
    cout << "Owner left" << endl;
    d1.closeDoor();
    h1.dayPassed();
    g1.gardening(h1, d1);
```

```
return 0;
}
```

Task

Based on the information provided in this manual, please improve the simple RPG character creation program.

Program requirements:

1. Add four classes to the program representing character professions (mage, warrior, berserker, thief)
2. Make the profession classes friends with a hero class.
3. Equip each profession class with a member function that increases the value of the corresponding hero's attribute (Mage - intelligence, warrior endurance, berserker - strength, thief - dexterity)