Advanced Programming Concepts with C++

CSI 2372

Tutorial # 6

Selected exercises from chapters 7 and 13

uOttawa

Ahmedou Jreivine

Exercise 7.52:

- Using our first version of `Sales_data` from § 2.6.1 (p. 72), explain the following initialization. Identify and fix any problems.
  - Sales_data item = {"978-0590353403", 25, 15.99};
- **Answer:**

```
#include <string>
struct Sales_data {
    std::string bookNo;
    unsigned units_sold; //= 0;
    double revenue; //= 0.0; (Error: Cannot provide in-class initializer for an
                                  aggregate class (since C++11) (until C++14))
};
int main() {
    Sales_data item = {"978-0590353403", 25, 15.99};
  return 0;
}
```

Exercise 7.56:

- What is a ***static*** class member? What are the advantages of ***static*** members? How do they differ from ordinary members?
- **Answer:**
  - A ***static*** class member is a member that is associated with the class, rather than with individual objects of the class type. It exists outside any object of the class.
  - **Advantages:**
    - Storage efficient.
    - If a static member of a class changes, each object of the class will use the new value of that static member.
    - A static data member can have incomplete type.
    - A static member (data or function) can be used as a default argument.
  - Differences between ordinary members and static members: A static member belongs to the class, an ordinary member belongs to objects of the class.

Exercise 7.57: Write your own version of the Account class.

```
#include <string>
#include <iostream>
class Account {
        static constexpr int period = 30;
        friend std::ostream &print(std::ostream &, const Account &);
public:
        Account() : Account("", 0) {}
      //explicit Account(const std::string &o) : Account(o, 0.0) {}
        explicit Account(const std::string &o, double b = 0.0)
                           : owner(o), balance(b), daily_tbl() {}
        void addInterest() { balance += balance * interestRate; }
        std::ostream &print(std::ostream &);
        static double getRate() { return interestRate; }
        static void setRate(double newRate) { interestRate = newRate; }
private:
        std::string owner;
        double balance;
        double daily_tbl[period]; //must be after the initialization of `period`.
        static double interestRate;
        static double initRate();
};
```

Exercise 7.57: Write your own version of the Account class.

```
double Account::interestRate = initRate();
double Account::initRate() {  return 0.1; }
constexpr int Account::period;
std::ostream &print(std::ostream &os, const Account &a) {
        os << a.owner << " " << a.balance << "\n"  << a.daily_tbl[0];
        return os;
}
int main() {
        Account act1;
        Account act2("Zhang San");
        Account act3("Li Si", 100.50);
        //Account act4 = "Wang";  // Error (implicit constructor call)
        print(std::cout, act1) << std::endl;
        print(std::cout, act2) << std::endl;
        print(std::cout, act3) << std::endl;
        return 0;
}
```

Exercise 7.58:

- Which, if any, of the following static data member declarations and definitions are errors? Explain why.

```
// example.h
class Example {
public:
    static double rate = 6.5;
    static const int vecSize = 20;
    static vector<double> vec(vecSize);
};
// example.C
#include "example.h"
double Example::rate;
vector<double> Example::vec;
```

Answer 7.58:

```
// example.h
class Example {
public:
        static double rate; // = 6.5;
        // static member should be initialize outside class
        static const int vecSize = 20;
        static vector<double> vec; //(vecSize);
        // 1. cannot use parentheses as in-class initializer
        // 2. static member should be initialize outside class
};

// example.C
#include "example.h"
double Example::rate = 6.5;
// should initialize static data member
vector<double> Example::vec(vecSize);
// should initialize static data member
```

Refereces

Accreditation:

- This presentation is prepared/extracted from the following resources:
  - C++ Primer, Fifth Edition.
    Stanley B. Lippman Josée Lajoie Barbara E. Moo
  - https://github.com/jaege/Cpp-Primer-5th-Exercises
  - https://github.com/Mooophy/Cpp-Primer