|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên học phần:** | **Phương pháp lập trình hướng đối tượng** | **Mã HP:** | **CSC10003** |  |
| Thời gian làm bài: | **100 phút** | Ngày thi: | **30/12/2019** |  |
| Ghi chú: *Sinh viên* [ 🞎 *được phép /* **☑** *không được phép*] *sử dụng tài liệu khi làm bài.* | | | |  |
|  | | | | |

**Họ tên sinh viên**: …............................................................. **MSSV**: …………… **STT**: …..

**Câu 1** (4 *điểm*)**.**

a) Phân biệt overload và override. Cho ví dụ minh họa. (1 điểm)

b) Đâu là điểm giống và khác nhau của hàm ảo (virtual) và hàm thuần ảo (pure virtual) (1 điểm)

c) Số Armstrong là số mà từng chữ số trong số đó lũy thừa với số chữ số của nó bằng chính nó.

Ví dụ: 153 = (1^3) + (5^3) + (3^3) = 1 + 125 + 27 = 153.

Bạn hãy khai báo lớp, viết chương trình để hàm main có thể chạy được mà không bị báo lỗi. Kết quả in ra màn hình là gì? Giải thích? (Không cần viết lại hàm main)

int main() {

CSoArmstrong a;

cin >> a; //người dùng sẽ nhập giá trị a bất kỳ, nếu a không phải

//là số Armstrong thì chuyển về số Armstrong gần nhất.

cout << a;

CSoArmstrong b(a);

a++; //tăng lên số Armstrong gần nhất.

a = ++b;

if(a != b)

cout<< "So " << a << “ khac “ << b;

else

cout<< "So " << a << “ giong “ << b;

return 0;

}

**Câu 2** (6 *điểm*)**.**

A screenshot of a cell phone screen with text

Description automatically generatedCho trước lớp ngày (Date) có thiết kế như sau:

* Hàm tạo không đối Date() sẽ trả ra ngày tháng hiện tại.

**Ngữ cảnh**

RaltaBank là tham vọng startup nhằm tạo ra ví điện tử.

Một khách hàng tại RaltaBank sẽ có các thông tin: Mã khách hàng(int), Họ và tên (string), Số điện thoại (string), Email (string), Giới tính (bool), Địa chỉ (string), Email (string), Ngày tháng năm sinh (Date), Ngày mở tài khoản (Date), Đang hiệu lực (bool).

Một tài khoản của khách hàng sẽ có các thông tin: Mã tài khoản (int), Mã khách hàng (int), Số dư (double).

Một giao dịch tại ngân hàng sẽ được lưu các thông tin: Mã khách hàng (int), Số tiền (double), Loại giao dịch (‘Ghi có’ - Deposit / ‘Ghi nợ’ – Debit. Có thể dùng kiểu string), Ghi chú (string), Ngày phát sinh giao dịch (Date).

+ Khi tài khoản nhận tiền, loại giao dịch là Ghi có. Ví dụ đang từ 100.000 nhận thêm 50.000 thì được gọi là ghi có 50.000

+ Khi tài khoản bị trừ tiền do mua sắm online hoặc chuyển tiền, loại giao dịch là Ghi nợ. Ví dụ đang từ 100.000 bị trừ 20.000 thì được gọi là ghi nợ 20.000. Chú ý từ debit được dịch là ghi nợ nhưng ở đây không có … nợ nào cả mà tiền trừ ngay lập tức.

**Yêu cầu 1: (2 điểm)**

Vẽ sơ đồ lớp biểu diễn 3 đối tượng: Khách hàng (customer), Tài khoản (Account), và Giao dịch (balance). Với mỗi thuộc tính, cần vẽ luôn cặp hàm getter và setter tương ứng.

**Yêu cầu 2**

Nếu khách hàng đã mở tài khoản từ 3 tháng trở lên sẽ được phép dùng dịch vụ Fast Cash, là dịch vụ cho phép mượn tiền tối đa 10 triệu để chi tiêu khi số dư còn 0 đồng. Định kì mỗi tháng, hệ thống của ngân hàng sẽ quét qua danh sách khách hàng trong hệ thống và thực hiện nâng cấp tài khoản.

1. Hãy hiệu chỉnh và vẽ lại sơ đồ lớp biểu diễn tài khoản để có thể biết được tài khoản này đã có thể dùng được tính năng Fast Cash. **(1 điểm)**
2. Hãy viết hàm nâng cấp định kì cho các khách hàng **(1 điểm),** biết nguyên mẫu hàm như sau

int CustomerBus::UpgradeFastCash(vector<Customer\*> customers);

Biết hàm với nguyên mẫu: void CustomerDao::Save(Customer\* customer)

sẽ chịu trách nhiệm lưu lại thông tin của một khách hàng vào cơ sở dữ liệu.

(Đối tượng thuộc lớp Customer\* chỉ chịu trách nhiệm chứa dữ liệu được nạp từ csdl lên bộ nhớ mà thôi)

Để lấy thông tin tài khoản từ thông tin của khách hàng, bên trong đối tượng Customer có một thuộc tính public tên là Account, có kiểu là Account để bạn có thể truy cập thông tin tài khoản tương ứng của một khách hàng.

**Yêu cầu 3 (2 điểm)**

Trong mỗi tháng, ngân hàng sẽ tiến hành bốc thăm trao thưởng cho khách hàng, mỗi người một chiếc iPhone 11 Pro cấu hình max. Điều kiện là nếu khách hàng có tháng sinh trùng với tháng hiện tại thì tích của ngày, tháng và năm tạo thành một số Armstrong.

Hãy cài đặt hàm bốc thăm này với nguyên mẫu như sau:

Customer\* CustomerBus::FindBirthDayWinner(vector<Customer\*> customers)