



Họ và tên SV: Nguyễn Văn Lộc
Mã số SV: 20120131
Ngày thi: 15/1/2022 Giờ thi: 09^h55

Tên học phần: Phương pháp lập trình hướng đối tượng
Mã học phần: CSC10003
Số trang/Tổng số trang: 1 / 9

Câu 1:

a. Kết quả mà đoạn chương trình trên in ra màn hình:

default

derive default

derive default

Trình tự hoạt động của chương trình.

+ Dòng 27: một đối tượng d của lớp Derive được tạo ra. Lớp Derive không được định nghĩa phương thức khởi tạo nào nên trình biên dịch sẽ cấp cho nó phương thức khởi tạo mặc định, gọi đến phương thức khởi tạo mặc định của lớp cơ sở (là lớp Base), do đó, m-ptr sẽ được gán là "default".

+ Dòng 28: bar (&d, d):

&d → foo(); // đây là con trỏ lớp kế thừa nên không có cơ chế đa hình

~~Do &d là con trỏ nên cơ chế đa hình sẽ được kích hoạt do phương thức foo() của lớp Base là hàm ảo. Khi đó, giá trị của m-ptr là "default" sẽ được in ra màn hình.~~

~~Sau~~ *, phương thức foo() của lớp Derive được gọi, in ra màn hình "derive", rồi lại gọi đến phương thức foo() của lớp Base bằng lệnh Base::foo(), in ra màn hình "default".

d. foo();

Do d chỉ không phải là con trỏ nên không kích hoạt được cơ chế đa hình. Khi đó, phương thức foo() của lớp Derive được gọi, in ra màn hình "derive", rồi gọi đến phương thức foo() của lớp Base, in ra màn hình "default".



Họ và tên SV: Nguyễn Văn Lộc
Mã số SV: 20120131
Ngày thi: 15/1/2022 Giờ thi: 09h55

Tên học phần: Phương pháp lập trình hướng đối tượng
Mã học phần: CSC 1000.3
Số trang Tổng số trang: 2 9

81. Viết lại lớp Base như sau:

```
class Base {
```

```
private:
```

```
    char* m_ptr;
```

```
public:
```

```
    Base() { m_ptr = strdup("default"); }
```

```
    Base(const Base& b) { m_ptr = strdup("copy"); }
```

```
    virtual void fo() { std::cout << m_ptr << "\n"; }
```

```
    Base& operator = (const Base& b) {
```

```
        if (m_ptr) {
```

```
            free(m_ptr);
```

```
            m_ptr = strdup(b.m_ptr);
```

```
        } else {
```

```
            m_ptr = strdup(b.m_ptr);
```

```
        }
```

```
    }
```

```
    ~Base() {
```

```
        free if (m_ptr) {
```

```
            free(m_ptr);
```

```
            m_ptr = NULL;
```

```
        }
```

```
    }
```

```
};
```

Ở đây, ta thêm vào phương thức hủy và toán tử gán cho lớp Base vì lớp Base có thuộc tính con trỏ được cấp phát bộ nhớ. Còn lớp

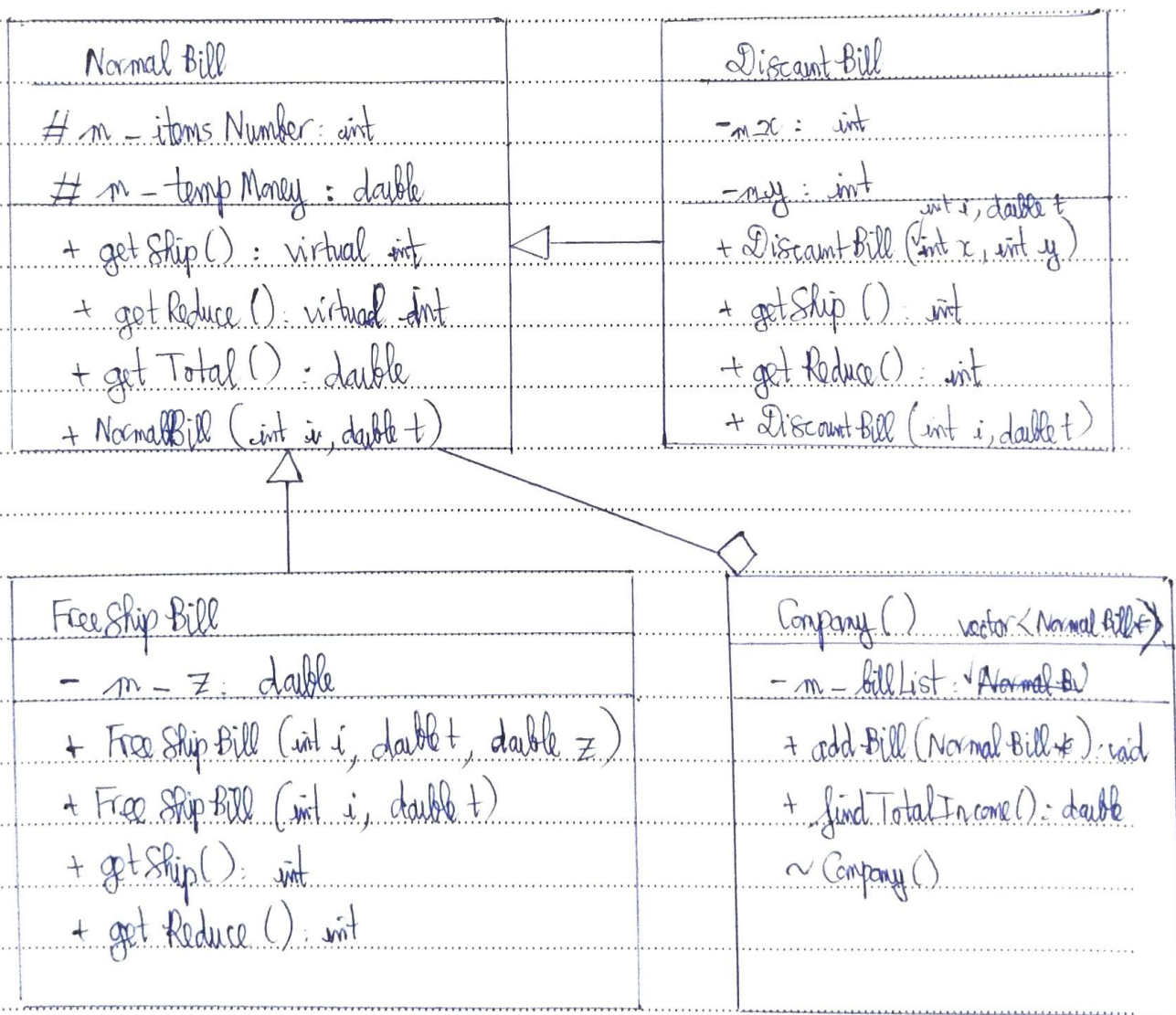
Họ và tên SV: Nguyễn Văn Lộc
Mã số SV: 20120431
Ngày thi: 15/1/2022 .. Giờ thi: 09h 55

Tên học phần: Phương pháp lập trình hướng đối tượng
Mã học phần: CSC1003
Số trang/Tổng số trang: 3 / 9

Derive không có thuộc tính con tạo cơ chế phát lại như nên ta không cần thêm.

Câu 2:

a)





Họ và tên SV: Nguyễn Văn Lộc
Mã số SV: 20120131
Ngày thi: 15/1/2022 .. Giờ thi: 09^h55

Tên học phần: Phương pháp lập trình hướng đối tượng
Mã học phần: CSC10003
Số trang/Tổng số trang: 4/9

8/

```
#include <iostream>
```

```
#include <vector>
```

```
const int DEFAULT_SHIP = 30; const int DEFAULT_X = 10; const int
```

```
const int DEFAULT_Y = 50; const int DEFAULT_Z = 1000;
```

```
using namespace std;
```

```
class Normal Bill {
```

```
private-protected: // làm việc protected để có thể sử dụng trong lớp kế thừa
```

```
int m - itemsNumber;
```

```
double m - tempMoney;
```

```
public:
```

```
Normal Bill (int i, double t) {
```

```
    m - itemsNumber = i;
```

```
    m - tempMoney = t;
```

```
}
```

```
virtual int getShip() {
```

```
    return DEFAULT_SHIP;
```

```
}
```

```
virtual int getReduce() {
```

```
    return 0;
```

```
}
```

```
double getTotal() {
```

```
    return 1.00 * (1 - 1.00 * getReduce() / 100) * tempMoney +
```

```
    1.00 * getShip();
```

```
}
```

```
};
```

Họ và tên SV: Nguyễn Văn Lộc
Mã số SV: 20120131
Ngày thi: 15/1/2022 Giờ thi: 09h 55

Tên học phần: Phương pháp lập trình hướng đối tượng
Mã học phần: CSC10003
Số trang Tổng số trang: 5 9

```
class DiscountBill : public NormalBill {
```

```
private:
```

```
    int m - x;
```

```
    int m - y;
```

```
public:
```

```
    DiscountBill (int i, double t, int x, int y) {
```

```
        DiscountBill (int i, double t, int x, int y) : NormalBill (i, t) {
```

```
            m - x = x;
```

```
            m - y = y;
```

```
        DiscountBill (int i, double t) : NormalBill (i, t) {
```

```
            m - x = DEFAULT - X;
```

```
            m - y = DEFAULT - Y;
```

```
        int getShip() {
```

```
            return NormalBill::getShip();
```

```
        int getReduce() {
```

```
            return NormalBill::
```

```
                if (m - itemsNumber > m - y) {
```

```
                    return NormalBill::getReduce() + x;
```

```
                return NormalBill::getReduce();
```

```
};
```




Họ và tên SV: Nguyễn Văn Lễ
Mã số SV: 20120131
Ngày thi: 15/1/2022... Giờ thi: 08^h 55

Tên học phần: Phương pháp lập trình hướng đối tượng
Mã học phần: CSC10003
Số trang/Tổng số trang: 6/9

```
class FreeShipBill : public NormalBill {  
private:  
    double m - z;  
public:  
    FreeShipBill (int i, double t, double z): NormalBill (i, t) {  
        m - z = z;  
    }  
    FreeShipBill (int i, double t): NormalBill (i, t) {  
        m - z = DEFAULT - z;  
    }  
    int getShip () {  
        int coefficient;  
        if (m - tempMoney >= m - z) {  
            coefficient = 0;  
        } else {  
            coefficient = 1;  
        }  
        return coefficient * NormalBill::getShip();  
    }  
    int getReduce () {  
        return NormalBill::getReduce();  
    }  
};
```

Họ và tên SV: Nguyễn Văn Lộc
Mã số SV: 20120131
Ngày thi: 15/1/2022. Giờ thi: 09h55

Tên học phần: Phương pháp lập trình hướng đối tượng
Mã học phần: CSC 10003
Số trang/Tổng số trang: 7/9

```
class Company {
```

```
private:
```

```
vector<NormalBill*> m_billList;
```

```
public:
```

```
void addBill(NormalBill* p) {
```

```
    m_billList.push_back(p);
```

```
}
```

```
double findTotalIncome() {
```

```
    double res = 0;
```

```
    for (int i = 0; i < m_billList.size(); i++) {
```

```
        res += m_billList[i] -> getTotal();
```

```
}
```

```
    return res;
```

```
}
```

```
~Company() {
```

```
    for (int i = 0; i < m_billList.size(); i++) {
```

```
        delete m_billList[i];
```

```
        m_billList[i] = nullptr;
```

```
}
```

```
}
```

```
};
```

Họ và tên SV: Nguyễn Văn Lộc
Mã số SV: 20120131
Ngày thi: 15/1/2022. Giờ thi: 09h55

Tên học phần: Phương pháp lập trình hướng đối tượng
Mã học phần: CSC10003
Số trang/Tổng số trang: 8/9

```
int main() {
```

```
    Company c;
```

```
    int n;
```

```
    cout << "Nhập số hàng đơn hàng: \n";
```

```
    cin >> n;
```

```
    cout << "Vui lòng nhập đơn hàng, \n";
```

```
    cout << "Nhập 0 nếu đây là đơn hàng bình thường \n";
```

```
    cout << "Nhập 1 nếu đây là đơn hàng giảm giá \n";
```

```
    cout << "Nhập 2 nếu đây là đơn hàng miễn phí giao hàng \n";
```

```
    cout << "Sau đó, nhập thông tin về mặt hàng đã chọn  
    và số tiền tạm tính \n";
```

```
    for (int i = 0; i < n; i++) {
```

```
        int type;
```

```
        cin >> type;
```

```
        int items;
```

```
        cin >> items;
```

```
        double money;
```

```
        cin >> money;
```

```
        if (type != NormalBill * pBill = nullptr;
```

```
        if (type == 0) {
```

```
            pBill = new NormalBill(items, money);
```

```
        } else if (type == 1) {
```

```
            pBill = new DiscountBill(items, money);
```

```
        }
```




Họ và tên SV: Nguyễn Văn Lộc
Mã số SV: 20120131
Ngày thi: 15/1/2022 Giờ thi: 09h55

Tên học phần: Phương pháp lập trình hướng đối tượng
Mã học phần: CSC10003
Số trang/Tổng số trang: 9/9

```
else if (type == 2) {
```

```
    pBill = new FreeShipBill(items, money);
```

```
    } else {
```

```
        cat << "Loại đơn hàng không hợp lệ";
```

```
    }
```

```
    c.addBill(pBill);
```

```
    }
```

```
    cat << "Tổng doanh thu của công ty là " <<
```

```
    c.findTotalIncome() << "\n";
```

```
    return 0;
```

```
    }
```