Câu 1: Cho đoạn mã nguồn sau: (Đã include đầy đủ các thư viện chuẩn cần thiết)

```
1  class Bike {
2  private:
3     char* brand; // hiệu xe
4  public:
5     virtual void move(int t1) {
6         cout << brand << ":" << t1*12 << " ";
7     }
8  };
9
10 class EBike : public Bike {
11     public: void move(int t2) {
12       Bike::move(t2*2);
13    }
14 };
```

```
15 void display(Bike& a, EBike& b) {
16    a.move(2);
17    b.move(2);
18 }
19
20 int main() {
21    EBike b1, b2;
22    display(b1, b2);
23    return 0;
24 }
```

a. Hãy cài đặt bổ sung các phương thức cần thiết để giải quyết các vấn đề về thuộc tính con trỏ cho lớp đối tượng Bike.
b. Cho biết kết quả xuất ra màn hình của chương trình trên (sau khi đã bổ sung code đầy đủ ở câu (a)). Giải thích ngắn gọn.
c. Nêu sự khác biệt giữa *lớp đối tượng (class)* và *đối tượng cụ thể (object, instance)*. Liệt kê các *lớp đối tượng* và các *đối tượng cụ thể* trong đoạn mã nguồn trên.

Câu 2: Cho đoạn mã nguồn sau: (Đã include đầy đủ các thư viện chuẩn cần thiết)

```
1  class IFace {
2  public:
3     virtual void show()=0;
4      virtual IFace* clone()=0;
5     virtual ~IFace() {
6     }
7  };
```

```
8  class Face : public IFace {
9  private:
10    string shape;
11 protected:
12    string getShape();
13 public:
14    Face(string sh) : shape(sh) {
15    }
16    virtual void show() {
17       cout << “Shape: ” << shape << endl;
18    }
19 };
```

a. Hãy viết lớp `EyedFace` có:
   - Hai thuộc tính: `shape` kiểu chuỗi và `eyes` kiểu số nguyên;
   - Hai phương thức: `show()` xuất ra màn hình giá trị của cả 2 thuộc tính, và `clone()` trả về một đối tượng `EyedFace` mới là một bản sao của đối tượng này;
   - Một hàm tạo (constructor) nhận 2 tham số tương ứng với 2 thuộc tính.

   Yêu cầu: Tái sử dụng tối đa mã nguồn đã cho.

b. Cho đoạn mã nguồn sau nối tiếp theo phần trên (cả đoạn cho sẵn lẫn của câu (a)):

```
1  void testFace(IFace* fc) {
2      IFace* a[3] = { fc, fc->clone(), fc->clone() };
3      for(int i=0; i<3; i++) {
4        a[i]->show();
5      }
6      cout << "The same 3 lines?";
7  }
```

```
8  int main() {
9     Face fc;
10    Face fc1("Rectangle");
11    testFace(&fc1);
12    return 0;
13 }
```

- Hàm `main()`có lỗi không thể chạy được. Hãy giải thích cụ thể lý do lỗi đó.
- Hãy sửa lỗi của hàm `main()`để chạy được hàm `testFace()` và cho biết kết quả xuất ra màn hình của hàm `main()`đó.

c. Hãy cải tiến chương trình bên trên (cả 2 câu (a) và (b)) như sau:

- Hàm `testFace()`quản lý bộ nhớ chưa hiệu quả. Hãy sửa lại lỗi bộ nhớ đó.
- Hãy viết thêm mã vào lớp `EyedFace` để đếm tổng số đối tượng (object, instance) của lớp đó và cập nhật mỗi khi tạo/huỷ đối tượng. Sau đó hãy thêm mã vào cuối hàm `main()`để kiểm tra xem còn bao nhiêu đối tượng `EyedFace` trong bộ nhớ.
  Lưu ý: Chỉ cần ghi phần mã bổ sung và tên lớp/hàm, còn lại thì ghi ba chấm.

Câu 3:

Một công ty viễn thông cung cấp dịch vụ điện thoại và internet hỗn hợp cho khách hàng theo hình thức trả sau. Để sử dụng dịch vụ, khách hàng cần ký hợp đồng với công ty. Trong hợp đồng, cần có các thông tin cá nhân của khách hàng (họ tên, chứng minh nhân dân, địa chỉ) và thông tin về cách tính cước mà khách hàng chọn. Cuối mỗi tháng, khách hàng sẽ được thông báo cước tùy theo lượng sử dụng của mình tương ứng với gói cước đã đăng ký ban đầu.

Hợp đồng với gói cước Basic có cách tính tiền như sau:

Cước điện thoại = *Thời gian gọi* (phút) * *Đơn giá gọi* (1000 đồng/phút).

Cước internet = *Lưu lượng truy cập* (MB) * *Đơn giá truy cập* (200 đồng/MB).

Cước tổng = *Cước điện thoại* + *Cước internet* + 10% VAT.

Để thu hút thêm đối tượng khách hàng thường xuyên sử dụng internet, công ty mở rộng thêm hai loại hợp đồng mới có cách tính cước linh hoạt như sau:

Gói cước	**Cước điện thoại**	**Cước internet**
Data Free	Tương tự gói Basic	• Nếu *Lưu lượng truy cập* ≤ *Ngưỡng lưu lượng miễn phí* ⇨ Chỉ đóng *Cước thuê bao*. • Nếu *Lưu lượng truy cập* > *Ngưỡng lưu lượng miễn phí* ⇨ *Cước thuê bao* + *Cước lưu lượng vượt ngưỡng*. Ghi chú: - *Cước thuê bao* và *Ngưỡng lưu lượng miễn phí* được công ty xác định lúc lập hợp đồng đăng ký cho khách hàng và được ghi trong mỗi hợp đồng: Có thể khác nhau tuỳ vào lúc lập hợp đồng nhưng không đổi sau đó. - *Cước lưu lượng vượt ngưỡng* tính theo công thức *Cước internet* của gói Basic.
Data Fix	Tương tự gói Basic + Giảm 10% giá cước	Mức cố định 1.000.000 đồng

Hãy vẽ sơ đồ lớp và viết chương trình cho phép công ty quản lý các hợp đồng trong một danh sách duy nhất với 2 chức năng: cho phép khách hàng đăng ký hợp đồng mới và thông báo tiền cước cho tất cả khách hàng vào cuối tháng.

Yêu cầu:

- Áp dụng kế thừa và đa hình để tăng tính tái sử dụng và tính dễ mở rộng của chương trình.
- Sơ đồ lớp: thể hiện các lớp đối tượng, quan hệ giữa các lớp, các thuộc tính, các hàm trong mỗi lớp.
- Mã nguồn: ngôn ngữ C++.