سوال۱)

همانطور که می‌دانیم دسترسی به اعضای داده‌ای protected و private در خارج از کلاس امکان پذیر نمیباشد؛ با این حال تفاوت اصلی بین این دو نوع عضو داده‌ای در کلاس های فرزند نمایان می‌شود؛ بدین صورت ک اعضای protected در کلاس وارث قابل ویرایش، خواندن و ... میباشند اما اعضای privat همچنان بصورت hidden هستند.

در مثال انتخاب شده فرض کنید یک فروشگاه اینترنتی داریم. در این فروشگاه دو کاربر قابل تعریف هستند: یکی فروشنده و دیگری خریدار؛ که با توجه به نوع تعریف کاربر، هرکدام سطح دسترسی متفاوتی خواهند داشت. همچنین این سایت یک ادمین نیز دارد که بالاترین سطح دسترسی را داراست.

برای شبیه سازی این مثال، یک کلاس پدر با عنوان User تعریف کرده‌ایم؛ همانطور که مشاهده می‌شود این کلاس دارای چند عضو protected و چند عضو private است؛

```
class User
{
private:
    int _id;
    static int id;
    void showInfo()
    {
        cout << "SHOWING INFO FOR Id=" << _id << ":\t{Name: " << name << ",\tFamily: " << family << ",\t";
        switch (accessCode)
        {
        case 2:
            cout << username << " is a seller!}" << endl;
            break;
        case 3:
            cout << username << " is a buyer!}" << endl;
            break;
        }
    }
protected:
    string name, family, username, password;
    int accessCode;
public:
    User()
    {
        id++;
        _id = id;
    }
    friend class Admin;
};
```

در این مثال برای هر کاربرِ تعریف شده در سایت، بصورت خودکار یک ID منحصر به فرد تخصیص داده میشود.(ممکن است از این ID برای دریافت اطلاعات از سمت دیتابیس و ... استفاده شود) این عضو کلاس User برای هیچ کدام از دو کاربر تعریف شده (فروشنده و یا خریدار) در دسترس نباید باشد، پس تعریف آن بصورت private انتخاب درستی خواهد بود.

از طرفی دیگر عضوهایی داریم که برای کاربر باید دسترس‌پذیر باشد(مانند نام، یوزرنیم و پسورد و ...)؛ که از آنها میتوان برای Login و ... استفاده کرد. همچنین این اعضا توسط کلاس فروشنده و خریدار تعریف میشوند(لازم به ذکر است که این دو کلاس، از کلاس پایه یعنی User ارث بری میکنند)، که تعریف این اعضا بصورت protected این امکان را به ما میدهد تا کاربر به آنها دسترسی داشته باشد.

```cpp
class Seller : public User
{
public:
    Seller(string n, string f, string u, string p)
    {
        name = n;
        family = f;
        username = u;
        password = p;
        accessCode = 2;
    }
    void login(string u, string p)
    {
        if (u == username && p == password)
        {
            cout << "Logged in!" << endl;
        }
        else
        {
            cout << "Something went wrong!" << endl;
        }
    }
};
class Buyer : public User
{
public:
    Buyer(string n, string f, string u, string p)
    {
        name = n;
        family = f;
        username = u;
        password = p;
        accessCode = 3;
    }
    void login(string u, string p)
    {
        if (u == username && p == password)
        {
            cout << "Logged in!" << endl;
        }
        else
        {
            cout << "Something went wrong!" << endl;
        }
    }
};
```

همچنین یک کلاس به نام ادمین تعریف شده است که کلاسی دوست برای کلاس User است.

همانطور که گفته شد ادمین بالاترین سطح دسترسی را دارا است بنابراین می‌تواند به اطلاعاتی نظیر فعالیت هر کاربر و ... دسترسی داشته باشد.

(که این امر با فراخوانی ID مختص به آن کاربر امکان پذیر می‌باشد)

```
class Admin
{
protected:
    bool s = false;
    const string username = "Admin";
    const string password = "0000";
    int accessCode = 1;
public:
    void getUser(User ob)
    {
        if (s == true)
        {
            ob.showInfo();
        }
        else
        {
            cout << "You don't have permission to access this method!!!" << endl;
        }
    }
    Admin(string u, string p)
    {
        if (u == username && p == password)
        {
            cout << "Hello Admin!" << endl << "-----------" << endl;
            s = true;
        }
        else
        {
            cout << "Incorrect password or username!!!" << endl;
        }
    }
};
```