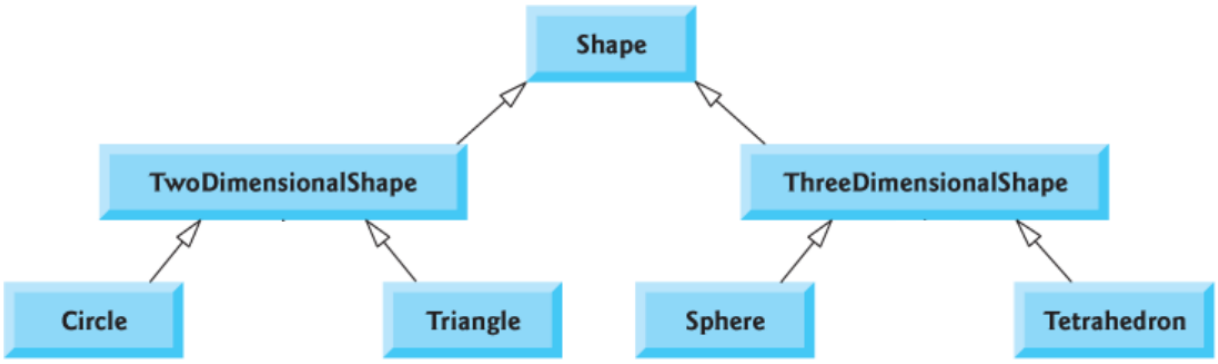
بسم الله الرحمن الرحیم

دانشگاه صنعتی شاهرود، دانشکده برق

درس برنامه نویسی پیشرفته

تمرین سری چهارم – موعد تحویل ۱۳ فروردین

۱. چند ریختی – کلاس شکل



- باتوجه به دیاگرام وراثت فوق، کلاسهای مناسب را تولید کنید. الگوی کلاس Shape را به صورت زیر در نظر بگیرید. این کلاس یک متغیر a دارد که بیانگر عرض (یا شعاع) است.

- برای کلاسهایی که مشتق می کنید به تناسب متغیرهای مناسب را اضافه کرده و تابع print را اصلاح کنید طوری که مشخصات شکل و مساحت (برای شکلهای دو بعدی) و حجم (برای شکلهای سه بعدی) را نشان دهد. توابع set و get مناسب را هم بنویسید.

- یک بردار vector از نوع Shape* ایجاد کنید و ۴ شیء مختلف از کلاسهای دایره، مثلث، کره و چهاروجهی، توسط آن ایجاد کرده و مقداردهی کنید. سپس در یک حلقه توابع print این اشیاء را فراخوانی کنید. جهت راهنمایی از تصویر زیر که مثالی مشابه از فصل ۱۲ کتاب (ویرایش ۱۰ ص ۱۱۳۴) است کمک بگیرید:

```
// create and initialize vector of three base-class pointers
vector<Employee*> employees{
   new SalariedEmployee("John", "Smith", "111-11-1111", 800),
   new CommissionEmployee("Sue", "Jones", "333-33-3333", 10000, .06),
   new BasePlusCommissionEmployee(
      "Bob", "Lewis", "444-44-4444", 5000, .04, 300)};
```

۲. چند ریختی – کلاس سازمان

- در یک شرکت دو نوع کارمند مشغول هستند. نوع اول، حقوق ساعتی دریافت می‌کنند. این دسته، نرخ دستمزد ثابتی برای هر ساعت دارند و حقوق آنها حاصل‌ضرب این نرخ در تعداد ساعت کار آنها می‌باشد. نوع دوم، حقوق ثابت دارند، به این ترتیب که برای ۱۴۰ ساعت کارکرد خود در ماه، مبلغ ثابتی دریافت می‌کنند و برای بیشتر از ۱۴۰ ساعت، به ازای هر ساعت ۵۰٪ اضافه دریافت می‌کنند. به عنوان مثال، اگر حقوق ثابت کارمندی ۱۴ میلیون ریال باشد، یعنی به ازای هر ساعت صدهزار ریال دریافت می‌کند. حال اگر چنین فردی ۱۸۰ ساعت کار کند، برای ۱۴۰ ساعت اول ۱۴ میلیون ریال و برای ۴۰ ساعت اضافه‌کار، ساعتی صد و پنجاه هزار ریال دریافت می‌کند. در نتیجه کل درآمد این شخص برای ۱۸۰ ساعت بیست میلیون ریال خواهد بود. یک کلاس مجرد کارمند تعریف

کنید که دو زیرکلاس برای دو نوع کارمند ساعتی و ثابت داشته باشد. تابع int earnings(int hours) را برای این کلاس‌ها تعریف کنید. فیلدها، سازنده‌ها و متدهای مورد نیاز را نیز برای این کلاس‌ها تعریف کنید .

- در ادامه‌ی سؤال قبل، کلاسی به نام Organization (سازمان) تعریف کنید که تعدادی کارمند (از هر دو نوع) را دارد، تابعی به شکل int total_earnings(int avg_hrs) برای کلاس سازمان تعریف کنید که مجموع درآمد تمام افراد سازمان را برمی‌گرداند اگر تمام کارمندها، به طور مساوی، avg_hrs ساعت در یک ماه کار کرده باشند.

- **اختیاری:** اگر کلاس سازمان شامل دو زیرسازمان از نوع Organization باشد، بخش قبل را طوری اصلاح کنید که درآمد تمام کارمندهای خود سازمان و زیرسازمانها را حساب کند.

۳. آشنایی بیشتر با وراثت

- این کلاس‌ها را درنظر بگیرید:

```
class A {
public:
	void f() { cout << "A::f"; }
	virtual void g() { cout << "A::g"; }
	void h() { cout << "A::h"; f(); g(); }
};

class B : public A {
public:
	void f() { cout << "B::f"; }
	void g() { cout << "B::g"; }
	void h() { cout << "B::h"; f(); g(); }
};
```

نتیجه‌ی قطعه کد زیر را (بدون اجرا کردن) مشخص کنید:

```
A a;
B b;
A* ap = &b;
ap->f();
ap->g();
ap->h();
```

سوالات اختیاری (۲۵ درصد نمره اضافه)

۴. آشنایی بیشتر با سربارگذاری عملگرها برای یک کلاس

- با مراجعه به فصل ۱۰ کتاب C++ how to program 10th.edition برنامه موجود در Fig. 10.3 تا Fig. 10.5 را پیاده سازی و اجرا کنید. به نظر شما چرا تابع مربوط به عملگرهای << و >> عضو کلاس معرفی نشده و به صورت تابع friend تعریف شده است؟ کدها را در این آدرس ببینید.

```cpp
// Fig. 10.3: PhoneNumber.h
// PhoneNumber class definition
#ifndef PHONENUMBER_H
#define PHONENUMBER_H

#include <iostream>
#include <string>

class PhoneNumber {
   friend std::ostream& operator<<(std::ostream&, const PhoneNumber&);
   friend std::istream& operator>>(std::istream&, PhoneNumber&);
private:
   std::string areaCode; // 3-digit area code
   std::string exchange; // 3-digit exchange
   std::string line; // 4-digit line
};

#endif
```

---

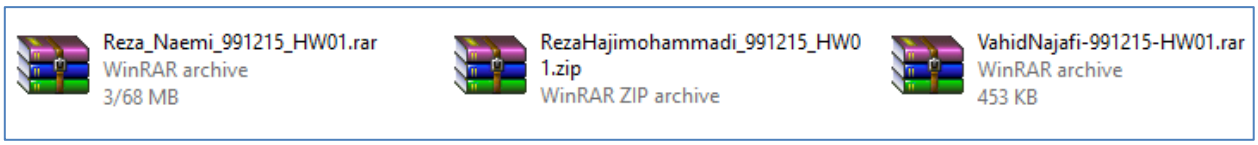
۵. خطاهای کامپایل برنامه‌ی زیر را (بدون کامپایل کردن) مشخص کنید:

```cpp
#include <iostream>
using namespace std;

class A {
public:
	void f() { cout << "A::f"; }
	virtual void g() = 0;
	void h() { cout << "A::h"; f(); g(); }
private:
	int a;
	void k() { cout << "A::k"; }
};

class B : public A {
public:
	int f() { cout << "B::f"; return 0; }
	int f(int x) { return x + a; }
	int k() { cout << "B::k"; }
protected:
	void h() { cout << "B::h"; }
};

class C : public A {
public:
	void g() { cout << "C::g"; }
};

int main() {
	A a;
	B b;
	C c;
	A* ap = &c;
	ap->g();
	B* bp = &c;
```

```
        bp->g();
}
```

نحوه ارسال تمرین

- **یک فایل word ایجاد کنید** و پس از ذکر نام و شماره دانشجویی و شماره تمرین، کد نوشته شده به همراه یک نمونه اجرای برنامه را در آن قرار دهید (از خروجی برنامه با فشردن همزمان دکمه‌های Alt و Print Screen عکس بگیرید و داخل فایل word کپی کنید).
- فایلهای حاوی کد و فایل پروژه (پسوندهای *.cpp, *.h, *.vcxproj) را به همراه فایل word، زیپ کنید. دقت کنید که پوشه های debug و release و فایل اجرایی برنامه (exe) و نیز فایلهای حجیم دیگر مثل *.db یا *.sdf و ... را **انتخاب نکنید**.
- عنوان فایل باید شامل **اسم خودتان**، **تاریخ ارسال** و **شماره تمرین** باشد مثلا VahidNajafi-991215-HW01.rar



- فایل زیپ شده را در سامانه LMS ارسال کنید.

نکته مهم: بخشی از نمره تمرین مربوط به تهیه فایل word به صورت خواسته شده، است.

تاخیر تا ۱ روز قابل اغماض است، تا ۵ روز، ۲۵٪ کسر نمره، بیش از ۵ روز، پذیرفته نخواهد شد

پیامبر اکرم صلی الله علیه وآله

ما عَمِلَ ابْنُ آدَمَ شَيْئاً أَفْضَلَ مِنَ الصَّلاةِ وَ صَلاحِ ذاتِ الْبَيْنِ وَ خُلُقٍ حَسَنٍ

انسان هیچ کاری بهتر از نماز، اصلاح میان مردم و خوش اخلاقی نکرده است.

نهج الفصاحه

موفق باشید؛ حسین خسروی