

به نام خداوند بخشنده‌ی مهربان

برنامه سازی پیشرفته

تمارین سری چهارم

مبحث: شیء گرایی – اشاره گر – اشاره گر به کلاس

مهلت تحویل: شنبه (۱۳۹۴/۱۲/۸)

نحوه‌ی تحویل:

لطفا تنها فایل های `cpp` هر سوال را (جمعا ۳ فایل به نام های `1.cpp` تا `3.cpp`) در یک پوشه تحت عنوان شماره دانشجوییتان قرار دهید و سپس زیپ کرده به آدرس زیر ایمیل کنید.

cpp1395+seri4@gmail.com

صرفا جهت اطلاع:

- حل تمارین انفرادی است. تمارین مشابه به اندازه ی نمره ی این تمارین برای طرفین نمره منفی دربر خواهد داشت.
- تمارین پس از موعد مقرر شده با کسر درصدی از نمره تا یک هفته فرصت دارد.
- امکان پرسش تصادفی تمارین از دانشجویان وجود دارد.

سوال (۱) توسعه کلاس `date`

لطفا اسلایدهای جلسات ۷ و ۸ را مرور کنید.

پیش فرض این سوال، تسلط شما به کلاس `date` تشریح شده در جلسه‌ی هشتم می‌باشد.

بدین منظور میتوانید کد

`"9_date_10.cpp"`

را از سایت دانلود کرده و این متدها را به آن اضافه کنید. سپس آن را با نام `1.cpp` ذخیره کرده و به عنوان پاسخ این سوال ارسال کنید.

در `main` برنامه از این متدها استفاده کنید.

الف) متد عضو کلاس `compare`

یک متد به نام `compare` به تاریخ اضافه کنید طوری که `d1.compare(d2)` در حالتی که $d1 < d2$ ، مقدار منفی یک، در حالتی که $d1 = d2$ ، مقدار صفر و در حالتی که $d1 > d2$ مقدار یک را برگرداند.

ب) بهبود متد `days_between`

در تعریف تابع `days_between` فرض شده است که پارامتر اول کوچکتر یا مساوی پارامتر دوم است. تعریف این تابع را طوری تغییر دهید که این محدودیت برداشته شود.

- به منظور خوانایی و همچنین قابلیت استفاده‌ی مجدد (`reusability`) از تابع عضو سوال ۱ استفاده کنید.

ج) متدی عضو کلاس `day_of_week`

با فرض این که ۱ فروردین ۱ (۱/۱/۱) جمعه است، متدی به کلاس تاریخ به نام `day_of_week` اضافه کنید که مشخص کند تاریخ مربوطه چه روزی از هفته است. مقدار که این تابع برمی‌گرداند یکی از رشته‌های `"sat, sun"` تا `"fri"` است.

سوال ۲) استفاده از کلاسی در کلاس دیگر

فایل راهنمای `"help_Triangle_Square.cpp"` را از سایت دانلود کنید.

این فایل حاوی دو کلاس مثلث و مربع است.

داخل این دو کلاس را تغییر ندهید و فقط از آن‌ها برای حل این سوال استفاده کنید.

بدین کد

`"help_Triangle_Square.cpp"`

را از سایت دانلود کرده و کلاس‌های مورد نیاز قسمت الف و ب را به آن اضافه کنید. سپس آن را با نام `2.cpp` ذخیره کرده و به عنوان پاسخ این سوال ارسال کنید.

راهنمایی: کلاسی که میخواهید از این دو کلاس استفاده کند باید بعد از این دو کلاس و قبل از `main` تعریف شود.

الف) کلاس `pentagon`

کلاسی به نام `pentagon` (پنج-ضلعی) بنویسید.

تابع سازنده:

- این کلاس باید تابع سازنده‌ای با دو مقدار `h` و `b` داشته باشد.

- این کلاس باید تابع سازنده‌ی پیش فرض داشته باشد (مقادیر دلخواه).

توابع عضو:

- تابع عضوی برای محاسبه‌ی مساحت این شکل بنویسید.

- تابع عضوی برای محاسبه‌ی محیط این شکل بنویسید.

پیاده سازی:

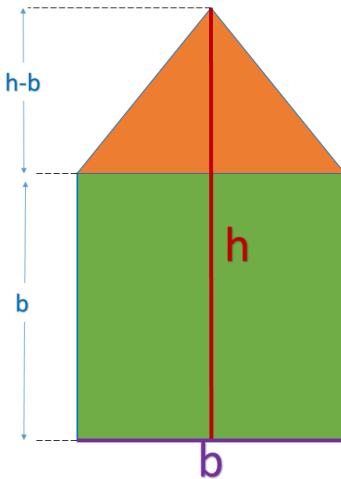
- این تابع باید شامل اعضای خصوصی (`private`) از انواع `Triangle` و `Square` باشد.

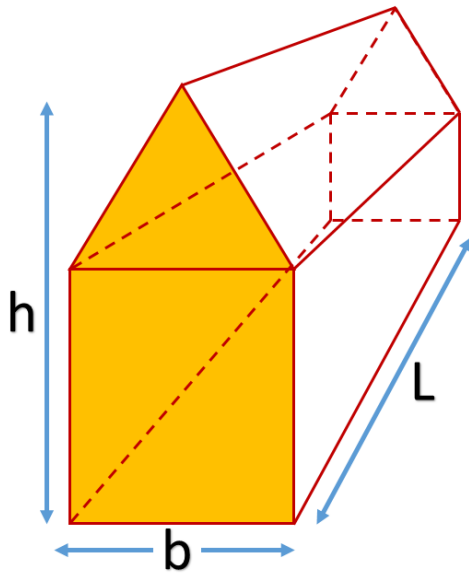
- تابع عضو `area` (محاسبه مساحت)، باید از تابع `area` موجود در دو شیء موجود دیگر کمک بگیرد.

- تابع عضو `perimeter` (محیط)، باید از تابع `perimeter` هر یک از دو شیء دیگر استفاده کند.

در `main` برنامه یک شیء از نوع این کلاس ساخته و مساحت و محیط آن را چاپ کنید.

محیط ۵ ضلعی مورد نظر = (محیط مربع + محیط مثلث) - یک ضلع مربع





ب) کلاس ThreeDPentagon

کلاسی به نام ThreeDPentagon تعریف کنید.

تابع سازنده:

این کلاس باید تابع سازنده‌ای با ۳ مقدار h ، b و L داشته باشد.

تابع عضو:

تابع عضوی برای محاسبه‌ی حجم بنویسید.

پیاده سازی:

این تابع باید از کلاس `pentagon` استفاده کند. این کلاس تنها دو عضو خصوصی دارد:

عددی از نوع `int` به نام L

شیء از نوع `pentagon`.

برای محاسبه‌ی حجم (`volume`) باید از تابع مساحت کلاس `pentagon` بهره ببرید.

در `main` برنامه یک شیء از نوع این کلاس ساخته و حجم آن را چاپ کنید.

سوال ۳) اشاره گری از نوع کلاس

لطفا کلاس `Square` سوال قبل را در فایل `3.cpp` کپی کنید.

حال در `main` برنامه با `new` کردن اشاره گری به نام `p` از نوع کلاس `Square` تعریف کنید. و مساحت آن را چاپ کنید.