گزارش تمرین چهارم

محبوبه شاکری

سوال اول)

Move semantics:

این متد در جاهایی که یک مقدار rvalue کپی میشود و از بین میرود به جای کپی شدن ، پوینتر های ان را عوض میکنیم تا از کپی شدن و کم شدن سرغت جلوگیری کنیم .

میتوانیم move constructor را برای کلاس بنویسیم تا کامپایلر در موارد مورد نیاز از این متد استفاده کند.

polymorphism:

به معنی چندریختی است و در کلاس هایی که چند کلاس دیگر با انواع مختلف از ان ارث میبرند کاربرد دارد . در این حالت تابعی که برای انواع مختلف تفاوت دارد به صورت virtual نوشته شده و در کلاس های مشتق شده تعریف میشود و در هنگام اجرای برنامه با توجه به نوع تابع مناسب اجر میشود.

pure abstract :

کلاس های Abstract کلاس هایی هستند که پیاده سازیشون به عهده برنامه نویس گذاشته شده ، یعنی برای متدهای اونها کدی نوشته نشده و برنامه نویس باید یک کلاس از اونها مشتق کنه و متدهای Abstractکلاس پدر رو درش پیاده کنه ، کلاس های Abstract یک جورهایی همون Interface هستند.

override :

اگر در یک کلاس، تابعی همنام تابعی در کلاسی که از ان ارث میبریم بنویسیم، overrideشده است. شرط overridingبه این صورت است که: توابع هم نام باشند، ورودی و خروجی هردو یکسان باشد و این دوتابع باید در دو کلاسی متفاوت باشد که یکی از دیگری ارث میربد

: Inline

:استفاده از این حالت برای توابع، باعث کاهش سرعت کامپایل می شود. با این کار کامپایلر کپی ای از تابع مورد نظر را در برنامه قرار داده و همانند دستورات دیگر، تابع را اجرا می کند. این حالت بیشتر در توابعی کاربرد دارد که دستورات آن کوتاه است و به صورت متعدد در چند جای برنامه فراخوانی شده است .

:explicit

به تبدیل هایی که به صورت آشکار توسط برنامه نویس با نوشتن نوع مورد نظر داخل برنامه مانند (cast_static)انجام می شود، تبدیل explicit میگویند.

اما تبدیل های implicit تبدیل هایی است که بصورت خودکار و مخفی انجام می شود .این موضوع هنگامی که کلاسی تعریف می کنیم جدی تر می شود. گاهی اوقات ممکن است ورودی کلاس نوعی متغیر است و ما ورودی را از جنس دیگری می دهیم. در این حالت ممکن است بصورت implicit و مخفی تبدیل انجام شود که در مواردی ما را دچار مشکل می کند .برای جلوگریی از این اتفاق کافی است توابع را به صورت explicit تعریف کنیم

سوال دوم)

در این سوال تفاوت capacity و size مشخص میشود ، در وکتور مقدار جای بیشتری در نظر گرفته میشود و هنگامی که size به ان میرسد capacity دوبرابر میشود . برای نشان دادن ان از تابع print که به صورت template تعریف شده تا وکتور با جنس های مختلف را بگیرد تعریف کرده و در یک حلقه از 1000 یک استرینگ تولید کرده و در پوینتر یونیک ان را در vector ، push می کنیم .

با دستور رزرو میتوانیم در ابتدا capacity وکتور خود را مشخص کنیم . در اینجا capacity در طول حلقه ثابت مانده و size ان عوض میشود .

در برنامه هایی که وکتور با size زیاد و مشخص داریم با reserve میتوانیم از کپی بهیوده جلوگیری کرده و سرعت را بهبهود ببخشیم.

سوال سوم)

کلاس های TwoDimensionalShape و ThreeDimensinalShapeاز کلاس Shape و کلاس های circleو square از کلاس دوبعدی و کلاس های sphere و cube از کلاس سه بعدی ارث بری می کنند .

کلاس Shape :از آنجایی که اکثر تابع ها در کلاس های پایینتر تعریف شده اند، در اینکلاس تابع های محدودی تعریف کرده ایم. یکی تابع print است که از نوع virtual و const تعریف کرده ایم، زیرا در توابع پایینتر نیاز داریم و همچنین در main از آن استفاده شده. و چون قرار نیست چیزی را تغییر دهد به صورت constتعریف شده است. متغیر double عدد پی را نیز تعریف کردیم که در محاسبات کلاس های پایینتر مورد استفاده قرار گرفت. در انتها اپراتور برای چاپ و cout را تعریف کردیم که از جنس ostreamاست و چون در خارج کلاس نیز استفاده می شود، آن را خارج از کلاس تعریف کردیم

کلاس های TwoDimensionalShape و ThreeDimensinalShape : در این کلاس ها. تابع print مانند کلاس Shape تعریف شده. در کلاس سه بعدی ها توابع area و volume و همچنین متغیر های x ، yو z را)همه از جنس(double تعریف شده است. در دو تابع area و volume چون کلاس های پایینتر هرکدام فرمول جداگانه دارند، مقداردهی اولیه را برابر صفر قرار دادیم .کلاس دوبعدی ها را هم مشابه کلاس سه بعدی نوشتیم، با این تفاوت که متغیر z و تابع volume موجود نیست .

کلاس های Sphere، Cube ، Square، Circle:در این کلاس ها ابتدا Constructor ها را نوشتیم که دایره و مربع 3 متغیر)شعاع یا ضلع، و مختصات مرکز(و کره و مکعب 4 متغیر را ورودی می گیرند. برای متغیر های مختصات، مقداردهی اولیه کردیم که اگر کاربر مختصاتی وارد نکرد، مختصات مرکز شکل،مبدا مختصات فرض شود. همچننی برای همه ی کلاس ها Constructor Copy هم نوشته شده است. در ادامه برای هر کلاس، طبق فرمول خاص خود، توابع area و برای شکل های سه بعدی volumeو همچنین تابع printنوشته شد.(همه virtual وconst) در انتهای هرکلاس هم اپراتور جمع نوشته شد، که ورودی اش از جنس کلاس point است. در این اپراتور مرکز مختصات ها با نقاط x و y و z نقطه جمع می شود .هرکدام از کلاس ها نیز دارای متغیر private از جنس double است که یا شعاع و یا ضلع را ذخیره می کند .

کلاس point :

در این کلاس متغیر های x, y , z به صورت private تعریف شده و سه تابع getx , gety ,getz برای دسترسی به ان ها تعریف شده است.

سوال چهارم)

برای تابع print حتما لازم است. زیرا در صورت سوال و قسمت main ،کلاس Shape باید پرینت شود و از آنجایی که هرکلاس، به صورت متفاوت باید چاپ شود، میخواهیم که این تابع به صورت جداگانه اجرا شود. اما برای قسمت area و volume بستگی دارد به اینکه در main از آن چطور استفاده شود. اگر هرتابع را داخل

خود کلاس های اشکال تعریف کنیم نیازی نیست. اما اگر این توابع را داخل کلاس باالتر)سه بعدی و دوبعدی(تعریف و از آنها به طور مستقیم استفاده کنیم، نیاز هست که به صورت virtual تعریف شود. درکد نوشته شده با اینکه نیازی نبود اما این توابع virtual تعریف شده اند. زیرا کلاس های دوبعدی و سه بعدی به طور مستقیم صدا زده نشدند

.به طور کلی در کلاس های ارث بری شده از virtual استفاده می کنیم. آن هم زمانی که توابعی در هر دو یا چند کلاس تعریف کرده ایم که هم اسم هستند و در قسمت های متفاوت می خواهیم از آنها استفاده کنیم. به عنوان مثال در این سوال تابع print برای همه ی کلاس های پایینتر تعریف شده است و میخواهیم که هرکدام اجرا شوند، اما ابتدا این تابع را برای کلاس بالاتر(Shape(استفاده کردیم.

سوال 5)

در این سوال میخواهیم کلاس stack را طوری پیاده سازی کنیم که توانایی ذخریه کلاس های مختلف را داشته باشد.

کلاس ctext :

تنها یک متغیر private از نوع string و constructor و تابع text_get را دارد که متغیر کلاس را برمیگرداند

کلاس stack :

این کلاس مشابه کلاس stack که توسط استاد نوشته شده است ، نوشته است با این تفاوت که به صورت template نوشته شده و چون template ای نوشته شده دارای فایل hpp است.

نحوه اجرای کد :

ابتدا یک text داخل کلاس CText داریم که سپس با استفاده از تابع push با char ها جمع میکنیم و داخل stack pushمی شود. سپس با استفاده از تابع pop آنها را خارج کرده و چاپ می کنیم .چون این کلاس صف از انتها پر شده و از انتها نیز تخلیه می شود، آخرین المان وارد شده، ابتدا نمایش داده می شود .در انتها نیز چون صف خایل شده است با پیغام "صف خایل است" مواجه می شویم

سوال 6)

الف)دستور remove مقادیر برابر با مقدار دلخواه ما را جا به جا میکند و iterator مربوط به انتهای مورد نظر ما را میدهد اما size را تغییر نمیدهد بنابراین با دستور erase از ان iterator تا انتها را پاک میکنیم تا به وکتور مد نظر برسیم .

ب) با دستور for_each مقدار هر خانه را دوبرابر میکنیم .

ج) در ابتدا میانگین را با کمک از accumulate دستور بدست اورده و یک شی از کلاس average ساخته و بر اساس ان وکتور را sort میکنیم.

ه) ابتوا وکتور را از مقدار کوچک به بزرگ sort می کنیم و باستقاده از unique و erase مقادیر تکراری را پاک می کنیم .

ر) ابتدا با دستور insert ، ابتدا و انتهای وکتور را وارد کرده و set با مقادیر وکتور پر میشوند . سپس iterator خانه ای که در ان 3 است را ذخیره کرده و با دستور erase ادامه را پاک میکنیم.

سوال 7) در کامنت ها توضیح داده شده است.

تمرینات در github به ادرس زیر قرار داده شده اند.

https://github.com/MahboobeSh/AP-HW4.git