دانشکده مهندسی برق

درس برنامه نویسی پیشرفته

استاد درس:دکتر جهانشاهی

تمرین سری چهارم

نام ونام خانوادگی:علیرضا خیاطیان

شماره دانشجویی:۹۵۲۳۰۳۹

آدرس github

https://github.com/alirezakhayyatian/AP-HW4

سوال اول

Move sematic : استفاده از متغیر های rvalueقبل از حذف شدن آن ها در واقع برای این که = operator را برای آرایه دینامیکی که از جنس lvalue باشد بنویسیم باید ارایه دینامیک جدیدی درست کنیم حال آنکه برای متغیر rvalue می توانیم از خود ارایه ورودی استفاده کرده و اصطلاحا آن را بدزدیم به این ترتیب کد بهینه می شود

Polymorphism: پلی مورفیسم کلمه به معنای داشتن اشکال مختلف است. به طور معمول، پلی مورفیسم زمانی رخ می دهد که سلسله مراتبی از طبقات وجود داشته باشد و آنها با ارث به هم مرتبط باشند.چند مورفیسم C ++ به این معنی است که فراخوانی به یک تابع عضو، بسته به نوع شی که تابع را فراخوانی میکند، تابع دیگری را اجرا میکند.

pure abstract :اگر مقدار تابعی را به صورت int volume() =0 تعریف کنیم کلاس مربوطهpure abstract شده و دیگر نمی توان object از این کلاس ساخت و حتما باید در کلاس هایی که از این کلاس ارث می برند این تابع را override کنیم

override :یعنی حتما تابعی به این شکل در کلاس مرجع هست که ما اینجا override می کنیم و به این ترتیب اطمینان حاصل میکنیم که کلاس مورد نظر را نوشته ایم

Inline: توابع درون خطی یک ویژگی پیشرفته C ++ برای افزایش زمان اجرای یک برنامه است. توابع می توانند به کامپایلر دستور داده شوند تا آنها را درون خطی ایجاد کنند تا کامپایلر بتواند آن تعریف تابع را جایگزین هر جا که نامیده می شود جایگزین کند. کامپایلر تعریف توابع درونی را در زمان کامپایل به جای تعریف تابع تعریف در زمان اجرا جایگزین می کند.

Explicit : برای constructor استفاده می شود تا به صورت ضمنی تبدیل فغحث نشود. به عنوان مثال، اگر شما یک کلاس با دو constructor یکی برای int و دیکری stringداشته باشید constructor ها هر کدام در جای خود call شوند

سوال دوم

یک وکتور از جنس unique_ptr از جنس string می سازیم که اشارگر از جنس *string است سپس در یک حلقه با ایجاد متغیر دینامیک و ذخیره آن در unique_ptr ادامه می دهیم برای push کردن unique_ptr در وکتور باید آن را rvalue کنیم برای این کار از move استفاده می کنیم .تابع printSize را جهت نمایش size و capacity ایجاد می کنیم

مشاهدا می شود مقدار capacity به صورت توان دوم افزایش پیدا می کند به این صورت که اگر المان ۱۷ اضافه شد capacity ۳۲ می شود و این ظرفیت تا زمانی که المان ۳۳ اضافه شود ثابت است ودر هنگام ظرفیت ۶۴ می شود

حال اگر وکتور را reserve کنیم به اندازه نزدیک ترین توان از دو ظرفیت میگیرد و با هر اضافه کرده مقدار capacity ثابت است و مقدار size نیز برابر مقدار داده شده است همچنین برای اضافه کردن عدد به انتهای عبارت Str تابع num را نوشته که صفر پشت عدد را نیز بدهد.در پایان برای بررسی صحت کد محتوای وکتور چاپ شده است

سوال سوم

ابتدا کلاس shape را تعریف می کنیم این کلاس pure abstract است و شی ایی از آن نمی توان ساخت دو کلاس twodimensionshape و threedimensionshape از آن ارث می برند و کلاس های circle & squre از اولی و cube & sphere تز دومی ارث می برند برای آن که operator<< را برای کلاس shape داخل کلاس تعریف کنیم آن را friend کرده و به این ترتیب آن را داخل کلاس می نویسیم در داخل این operator تابع print فراخوانی می کنیم این تابع را در کلاس shape به صورت abstract تعریف کرده و در کلاس دو بعدی سه بعدی نیز تعریف کرده به علاوه در هر یک از ۴ کلاس باقیمانده که هر یک برای یک شکل خاص است تعریف می کنیم حال برای اینکه با توجه به محتوای Shape* و نه نوع آن این تابع عمل کند در تمامی جاهایی که تعریف کرده ایم باید به صورت virtual تعریف کنیم به علاوه constructor های لازم را برای هر یک از ۴ شکل تعریف می کنیم که اگر بدون مختصات باشد نیز برنامه عمل کند اما برای محاسبه حجم و مساحت از دو تابع area و volume استفاده می کنیم که در سوال بعد به توضیح آن می پردازیم

کلاسpoint را پیاده سازی می کنیم به این طریق که مختصات دو بعدی و سه بعدی را دریافت کند و توسط تابع show چاپ کند حال از این کلاس در operator+ استفاده می کنیم operator+ را همانند print می کنیم. این تابع را در کلاس shape به صورت abstract تعریف کرده و در کلاس دو بعدی سه بعدی نیز تعریف کرده به علاوه در هر یک از ۴ کلاس باقیمانده که هر یک برای یک شکل خاص است تعریف می کنیم حال برای اینکه با توجه به محتوای Shape* و نه نوع آن این تابع عمل کند در تمامی جاهایی که تعریف کرده ایم باید به صورت virtual تعریف کنیم. حال متناسب با هر شکل اعم از دو بعدی و سه بعدی مختصات مرکز شکل را به اندازه point جا به جا می کنیم.برای اینکه تغیر را مشاهد کنیم یک بار دیگر ۴ شکل را چاپ می کنیم مشاهده می شود مرکز اشکال به اندازه نقطه داده شده جا به جا شده است.این بخش با عبارت after shifting از قسمت قبل تفکیک شده است.

سوال چهارم

توابع area و volume و print باید به صورت virtual باشند توابع را به صورت virtual تعریف می کنیم تا در مواردی که چند ریختی وجود دارد با توجه به محتوای Shape* و نه نوع آن این تابع عمل کند

سوال پنجم

توجه:کد این سوال به طور کامل زده شده است

ابتدا کلاس stack را به صورت template تعریف می کنیم میدانیم shared_ptr<Ctext> از جنس Ctext* می باشد که نیاز به حذف ندارد کلاس stack دارای یک وکتور template است در این برنامه template با shared_ptr<Ctext> جایگزین می شود این کلاس دو تابع getCount وisEmpty دارد که به ترتیب طول وکتور را میدهد و دیگری در صورتی که وکتور خالی باشد true برمی گرداندین کلاس دارای دو تابع push و pop است که به ترتیب عنصری از جنس template مذکور به وکتور اضافه یا کم می کند

کلاس CText نیز دارای یک متغیر string می باشد و دارای یک تابع getText است که این string را بر می گرداند

در واقع تابع pop در کلاس stack یک template از جنس shared_ptr<Ctext> بر میگرداند که با اعمال <- تابع getText از کلاس CText اجرا می شود..به علاوه در یک حلقه از A تا Z محتوای هر Ctext را به صورت دینامیک تولید کرده ودر shared_ptr<Ctext> ذخیره می کنیم.

سوال ششم

در ابتدا یک وکتور ایجاد می کنیم برای چاپ در هر مرحله ار each_for و تابع myfunction استفاده شده است

۱.دستور remove مقدار ۲ را از آرایه جمع کرده و به انتهای آن منتقل می کند و خروجی iterator ایی میدهد که انتهای وکتور بدون ۲ است برای حذف آن از erase استفاده بشود

۲.برای این که المان ها را در ۲ ضرب کنیم از دستور transform استفاده کرده و المان های جدید را نیز در همان بردار قبلی میریزیم و تابع mix را برای ضرب کردن نوشته ایم

۳.ابتدا با دستور accomulate میانگین را به دست آورده و سپس با کمک landa function نوشته شده بر اساس فاصله از میانگین sort می کنیم

۴.برای حذف عناصر تکراری ابتدا باید آن را مرتب کرد که در بخش قبل انجام شده است سپس از دستور unique که عناصر کنار هم را که با هم برابر هستند یکی از آن ها را حذف می کند استفاده می کنیم و در نهایت از دستور erase برای حذف اقدام می کنیم

۵.ابتدا با استفاده از دستور copy بردارا در set ذخیره می کنیم .برای حذف کردن اعداد بزرگ تر از ۳ از remove_if استفاده می کنیم اما چون این تابع برای set تعریف نشده ابتدا مقادیر set را در vector temp می ریزیم و با دستور remove_if در داخل دستور erase مقادیر بیشتر از ۳ را حذف می کنیم و در نهایت هم vectorرا همانند ابتدای سوال در set ذخیره می کنیم

سوال هفتم

تمامی containerها vector هستند .ابتدا یک بردار به ط.ل ۵۰ به نام a ایجاد می کنیم سپس بردار b را با مقادیر تصادفی ایجاد می کنیم برای این کار از دستور generate وتابع randGen ایجاد می کنیم

برای تولید بردار c ابتدا با استفاده از دستور iota وکتوری به طول ۵۰ با اعداد ۰ تا ۴۹ ایجاد می کنیم سپس جایگاه این اعداد در وکتور را به صورت رندوم و با استفاده از shuffle تعیین می کنیم و در نهایت آن را چاپ می کنیم به این ترتیب ارایه ایی از اعداد با فراوانی ۱ و به صورت رندوم ایجاد می کنیم.

روش دیگری که برای تولید بردار c در نظر گرفته ایم به این صورت است که ابتدا بردار c را ایجاد می کنیم برای این کار کانتینر را set انتخاب می کنیم تا بدین وسیله اعداد رندوم متمایز از یکدیگر باشند و تابع func را جهت تولید این set ایجاد کرده ایم

برای پیدا کردن توان دوم بردار های هر یک را با خودش ضرب داخلی می کنیم برای ضرب داخلی از دستور inner_product استفاده می کنیم