

تمرین های سری اول C++

- i. http://math.sharif.edu/~shahriari/Cpp_1.pdf
- ii. با استفاده از یک مثال که خودتان طرح کرده اید، ترتیب اجرای **constructor** و **destructor** را در سناریو های مختلف بررسی کنید. (طراح : رضوی)
- iii. تفاوت های میان **constructor** با **copy constructor** را توضیح دهید و با یک مثال که خودتان طراحی کرده اید تفاوت های آنها را از لحاظ **syntax** بررسی کنید. (طراح : رضوی)
- iv. می خواهیم یک درخت دودویی طراحی کنیم. طراحی کلاس **Node** با کلاس **BST** با شماست. درخت شما باید قابلیت نگه داری بیشمار **Node** را داشته باشد. کلاس **BST** می بایستی سه **constructor** داشته باشد. یکی از آنها که **default constructor** می باشد می بایستی درخت را برای حالت ابتدایی ۱۰۰ راس **initialize** کند. دیگری می بایستی با گرفتن مقدار دلخواه **n** این کار را انجام دهد. آخری نیز با گرفتن یک **vector** از **Node** ها، آنها را در درخت قرار می دهد. در کلاس **Node** می بایستی شماره **Node** و **Value** آن **Node** ذخیره شود. **BST** می بایستی که متد **insertNode** که یک **Node** یا **vector** ای از **Node** ها را می گیرد داشته باشد. پیاده سازی **deleteNode** که شماره **Node** را میگیرد و آنرا حذف می کند هم جایزه دارد. **BST** می بایستی متدهای **preOrder**, **inOrder**, **PostOrder** را داشته باشد و با فراخوانی آنها، **Node** هایی که به ترتیب پیمایش شده اند باید در کنسول چاپ شوند. (طراح : رضوی)

v. فرض کنید که از شما به عنوان یک برنامه نویس خواسته شده است که به مدیریت یک کارخانه در اداره ی امور کارخانه اش کمک کنید و یک برنامه ی ساده برای کنترل پایگاه های مختلف در کارخانه بنویسید. + ابتدا کلاسی با نام **Duty** بسازید که دارای خصوصیت های زیر باشد:

```
1. string name;
2. string mode =>{"IN", "OUT"};
3. double landa;
```

+ حال یک **class** با نام **WorkStation** طراحی کنید که در آن یک **object** از کلاس **Duty** را نگه میدارد، نگه داری وبه وجود آوردن این **object** به این صورت می باشد که کلاس **WorkStation** ابتدا از یک فایل با نام **processes.w#** (# شماره ایستگاه در کارخانه می باشد) دو **string** به ترتیب **name** و **mode** را می خواند و بعد یک **double** (هر سه این موارد هر کدام در یک خط جدا گانه در فایل نوشته شده اند). بعد شما میدانید یک برنامه نویس دیگر کلاسی با نام **Process** نوشته و این کلاس دارای متدی به شکل زیر است:

Void processes (Duty current_Duty);

این متد خصوصیات **current_Duty** را عوض می کند و بعد شما باید این خصوصیات را به ترتیبی که از فایل خوانده بودید در فایلی با نام **processes.w(#+1)** (اگر # برابر با 2 بوده باشد پس حال اطلاعات را در فایلی با اسم **processes.w3** بریزید).

+ اکنون کلاسی با نام **Factory** داریم که در آن به **n** تعداد **object** از کلاس **WorkStation** می سازیم و هر **object** کار خود را به ترتیب انجام می دهد (یعنی این کارها در یک حلقه قرار دارند و این حلقه زمانی تمام می شود که **WorkStation (1)** مشغول خواندن فایلی است خالی باشد). باید این نکته را ذکر کنم وقتی نوبت **WorkStation (n)**-ام است این کلاس اطلاعات جدید گرفته شده توسط متد درون کلاس **Processes** را در فایل **processes.1** می نویسد. (طراح : شهریار)

کسانی که این برنامه را به خوبی پیاده سازی کنند در اصل باید بدانند که یک نوع ساده از شبیه سازی رایانه ای انجام داده اند

نکات مهم

- i. سعی کنید از استناداری که در کلاس گفته شده پیروی کنید، پروتو تایپ کلاس در **header** به اسم کلاس و **definition** آن در فایل **cpp** به اسم همان کلاس نوشته شود. همه برنامه ها هم یک **main.cpp** دارند که تابع **main** در آن است و از کلاس ها استفاده می کند.
- ii. هرچا نیاز داشتید موردی را تعریف کنید ولی در صورت تمرین به آن اشاره نشده، با فرض این که این موارد شرایط خواسته شده در مسئله را عوض نمی کند، آزاد به انجام آن هستید.
- iii. از امکانات شی گرایی **C++** در مثال هایی که خوتان طراحی می کنید استفاده کنید.
- iv. برنامه ها بایستی بدون مشکل با کامپایلر استاندارد کامپایل شوند.
- v. تاریخ نهایی ارسال تمرین ها در کلاس حل تمرین اعلام می شود.
- vi. تمرین ها را به **TA** های ساعتی که به آن می روید بفرستید. آدرسهای آنها در زیر آمده است:

یک شنبه ها:

محسن خاکی mohsen.khaki@gmail.com

هومن مهاجری مقدم hooman67@gmail.com

چهار شنبه ها:

کاوه رضوی razavi1366@gmail.com

بهرز شهریاری shahriari13@yahoo.com