



آموزشگاه مجازی

هدف از این تمرین، آشنایی با multi-file programming می‌باشد. در این تمرین، شما باید پیاده‌سازی مربوط به یک آموزشگاه را انجام دهید. این آموزشگاه دروس مختلفی را ارائه می‌دهد. افراد در این آموزشگاه به دو دسته‌ی مدرس و دانش‌آموز تقسیم می‌شوند و از امکانات مختلف آن استفاده می‌کنند. شما باید برای برنامه‌ی خود، یک Makefile داشته باشید و از آن برای ایجاد فایل اجرایی نهایی خود استفاده کنید. همچنین، هر کلاس باید دارای یک فایل cpp و یک فایل h. جداگانه باشد. در غیر این صورت، نمره‌ای به آن تعلق نمی‌گیرد.

آموزشگاه

آموزشگاهی به تازگی شروع به کار کرده و از شما برای پیاده‌سازی امکانات اولیه مورد نیاز آن‌ها برای مدیریت درس‌ها و افراد، درخواست کمک کرده است. این آموزشگاه تعدادی مدرس و دانش‌آموز دارد و دروس مختلفی را ارائه می‌دهد و لیستی از آن‌ها را در خود ذخیره می‌کند. آموزشگاه لیستی از دروس ارائه شده و افراد دارد. توجه کنید که نام کاربری افراد و نام دروس یکتا فرض شده‌اند و لازم نیست این موضوع را بررسی کنید.

با دستور زیر، درس به آموزشگاه اضافه می‌شود.

```
course <course_name> <teacher_username>
```

دانش‌آموز، با دستور زیر به آموزشگاه اضافه می‌شود و دروس مشخص شده را برمی‌دارد. در این دستور فرض می‌شود که تمامی دروس مشخص شده حتماً از قبل به آموزشگاه اضافه شده‌اند و لازم نیست شما این موضوع را بررسی کنید. همچنین هر دانش‌آموز حداقل یک درس برمی‌دارد.

```
student <username> <course_name_1> <course_name_2> ... <course_name_n>
```

درس

هر درس، یک موضوع مشخص، یک مدرس و تعدادی دانش‌آموز دارد. با دستور زیر، اطلاعات درس قابل مشاهده است.

```
course_info <course_name>
```

ورودی و خروجی نمونه

ورودی	خروجی
course_info Python	Professor: Mr. X Students: A B C

مدرس

هر مدرس، یک نام کاربری یکتا و یک درس دارد.
هر مدرس با دستور زیر، می‌تواند لیست افرادی که درس او را دارند، مشاهده کند.

```
student_list <professor_username>
```

ورودی و خروجی نمونه

ورودی	خروجی
student_list Mr.X	A B C

دانش‌آموز

هر دانش‌آموز، یک نام کاربری یکتا و تعدادی درس دارد.
دانش‌آموز می‌تواند با دستور زیر، دروس خود را مشاهده کند.

```
course_list <student_username>
```

ورودی و خروجی نمونه

ورودی	خروجی
course_list A	Python JavaScript

نحوه‌ی تحویل

- فایل برنامه‌ی خود را با نام A10-SID.zip در صفحه‌ی CECM درس بارگذاری کنید که SID شماره‌ی دانشجویی شماست؛ این فایل باید شامل فایل‌هایی به شکل .cpp و .h و همچنین فایل مربوط به make باشد. برای مثال، اگر شماره‌ی دانشجویی شما ۸۱۰۱۱۲۳۴۵ است، نام فایل شما باید A10-810112345.zip باشد.
- لطفاً از روش‌های دیگر فشرده‌سازی مانند rar یا tar.gz استفاده نکنید.
- برنامه‌ی شما باید در سیستم عامل لینوکس و با مترجم g++ با استاندارد c++98 ترجمه و در زمان معقول برای ورودی‌های آزمون اجرا شود.
 - توجه کنید که هدف از این تمرین، آشنایی با multi-file programming است و در صورت multi-file نبودن برنامه، هیچ نمره‌ای به آن تعلق نمی‌گیرد.

- هر یک از کلاس‌های شما باید دارای یک فایل `.cpp` و یک فایل `.h` باشد و تعریف هیچ دو کلاسی نباید داخل یک فایل واحد قرار گرفته باشد.
- به فرمت و نام فایل‌های خود دقت کنید.
- از صحت فرمت ورودی‌ها و خروجی‌های برنامه‌ی خود مطمئن شوید.
- هدف این تمرین یادگیری شماست. لطفاً تمرین را خودتان انجام دهید. در صورت کشف تقلب مطابق قوانین درس با آن برخورد خواهد شد.
- برای آشنایی بیشتر با `multi-file programming` می‌توانید به لینک‌های زیر مراجعه کنید:

<https://www.mauriciopoppe.com/notes/os/bin/make/>

https://www.cs.swarthmore.edu/~newhall/unixhelp/howto_makefiles.html