



دانشگاه صنعتی شریف
دانشکده مهندسی کامپیوتر

برنامه درسی و ریز مواد دروس کارشناسی مهندسی کامپیوتر
نرم افزار

* مخصوص ورودی‌های ۱۳۸۸ به بعد *

نسخه بازنگری شده

دی ۱۳۹۰

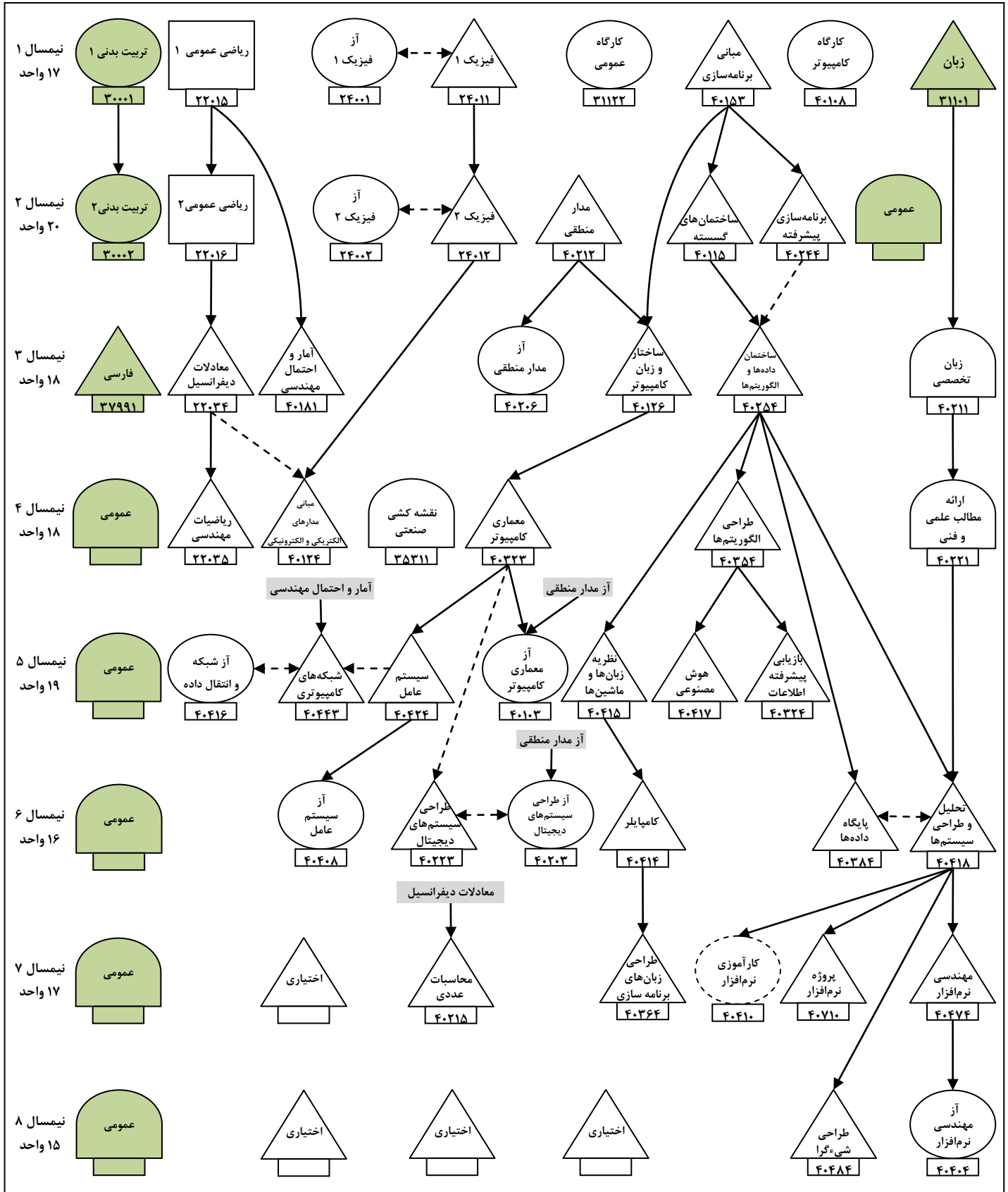
نمودار درسی و فهرست دروس

دوره کارشناسی مهندسی کامپیوتر - نرم افزار

* مخصوص ورودی‌های ۱۳۸۸ به بعد *

نمودار درسی دوره کارشناسی مهندسی کامپیوتر - نرم افزار

برای ورودی‌های ۸۸ به بعد (رعایت این برنامه برای ورودی‌های ۸۸ اختیاری، و برای ورودی‌های ۸۹ به بعد اجباری است)



پیش‌نیازی →

→ **هم‌نیازی**

تاریخ آخرین اصلاح: ۱۳۹۰/۶/۲

- دروس اختیاری گرایش نرم‌افزار**
- سیگنال‌ها و سیستم‌ها
 - گرافیک کامپیوتری
 - برنامه‌نویسی وب
 - مهندسی فناوری اطلاعات
 - آداب فناوری اطلاعات
 - امنیت شبکه
 - شبیه‌سازی کامپیوتری
 - توصیف و واریس برنامه‌ها
 - سیستم‌های خبره
 - مباحث پیشرفته در مهندسی نرم‌افزار

- مقررات دروس اختیاری**
- حداکثر یک درس کارشناسی ارشد دانشکده
 - حداکثر یک درس دیگر از گروه‌های دیگر دانشکده یا یک درس از دیگر دانشکده‌های دانشگاه (با تأیید مدیر گروه)
 - بقیه یا همه فقط از دروس مقابل

فهرست دروس عمومی دوره کارشناسی مهندسی کامپیوتر - نرم افزار (مجموعاً ۲۰ واحد)

ردیف	گرایش	واحد مورد نیاز	عنوان درس	واحد درس
۱	مبانی نظری اسلام	۴	اندیشه اسلامی ۱ (مبدأ و معاد)	۲
			اندیشه اسلامی ۲ (نبوت و امامت)	۲
			انسان در اسلام	۲
			حقوق اجتماعی و سیاسی در اسلام	۲
۲	اخلاق اسلامی	۲	فلسفه اخلاق (مباحث تربیتی)	۲
			اخلاق اسلامی (مبانی و مفاهیم)	۲
			آیین زندگی (اخلاق کاربردی)	۲
			عرفان عملی اسلام	۲
۳	انقلاب اسلامی	۲	انقلاب اسلامی ایران	۲
			آشنایی با قانون اساسی ج.ا.ایران	۲
			اندیشه سیاسی امام خمینی (ره)	۲
۴	تاریخ و تمدن اسلامی	۲	تاریخ فرهنگ و تمدن اسلامی	۲
			تاریخ تحلیلی صدر اسلام	۲
			تاریخ امامت	۲
۵	منابع اسلامی	۲	تفسیر موضوعی قرآن	۲
			تفسیر موضوعی نهج البلاغه	۲
۶	ادبیات	۳	ادبیات فارسی	۳
۷	زبان	۳	انگلیسی همگانی	۳
۸	تربیت بدنی	۲	تربیت بدنی ۱	۱
			تربیت بدنی ۲	۱
جمع واحدهای دروس عمومی که باید اخذ شود				۲۰

فهرست دروس پایه دوره کارشناسی مهندسی کامپیوتر - نرم افزار (مجموعاً ۲۵ واحد)

ردیف	عنوان درس	شماره درس	واحد
۱	ریاضی عمومی ۱	۲۲-۰۱۵	۴
۲	ریاضی عمومی ۲	۲۲-۰۱۶	۴
۳	معادلات دیفرانسیل	۲۲-۰۳۴	۳
۴	آمار و احتمال مهندسی	۲۲-۰۶۳	۳
۵	آزمایشگاه فیزیک ۱	۲۴-۰۰۱	۱
۶	آزمایشگاه فیزیک ۲	۲۴-۰۰۲	۱
۷	فیزیک ۱	۲۴-۰۱۱	۳
۸	فیزیک ۲	۲۴-۰۱۲	۳
۹	کارگاه عمومی	۳۳-۰۱۸	۱
۱۰	نقشه کشی صنعتی	۳۵-۳۱۱	۲
جمع واحدهای دروس پایه که باید اخذ شود			۲۵

فهرست دروس اصلی اجباری رشته مهندسی کامپیوتر - نرم افزار (مجموعاً ۸۳ واحد)

گروه‌های ۲ تا ۱۱ بر اساس حوزه‌های دانشی تعریف شده در آخرین نسخه برنامه پیشنهادی ACM/IEEE برای دوره کارشناسی نرم افزار (نسخه ۲۰۰۸) طبقه‌بندی شده‌اند.

۱- اصلی پایه

ردیف	نام درس	شماره درس	واحد
۱	ریاضیات مهندسی	۲۲-۰۳۵	۳
۲	مبانی مدارهای الکتریکی و الکترونیکی	۴۰-۱۲۴	۳
۳	زبان تخصصی	۴۰-۲۱۱	۲
۴	محاسبات عددی	۴۰-۲۱۵	۳
۵	ارائه مطالب علمی و فنی	۴۰-۲۲۱	۲
جمع واحدهای درسی که باید اخذ شود			۱۳

۲- ساختارهای گسسته

ردیف	نام درس	شماره درس	واحد
۱	ساختارهای گسسته	۴۰-۱۱۵	۳
جمع واحدهای درسی که باید اخذ شود			۳

۳- مبانی برنامه‌سازی

ردیف	نام درس	شماره درس	واحد
۱	کارگاه کامپیوتر	۴۰-۱۰۸	۱
۲	مبانی برنامه‌سازی	۴۰-۱۵۳	۳
۳	برنامه‌سازی پیشرفته	۴۰-۲۴۴	۳
۴	ساختار داده‌ها و الگوریتمها	۴۰-۲۵۴	۳
جمع واحدهای درسی که باید اخذ شود			۱۰

۴- الگوریتم‌ها و پیچیدگی

ردیف	نام درس	شماره درس	واحد
۱	طراحی الگوریتم‌ها	۴۰-۳۵۴	۳
۲	نظریه زبانها و ماشین‌ها	۴۰-۴۱۵	۳
جمع واحدهای درسی که باید اخذ شود			۶

۵- معماری و سازمان کامپیوتر

ردیف	نام درس	شماره درس	واحد
۱	آز معماری کامپیوتر	۴۰-۱۰۳	۱
۲	ساختار و زبان کامپیوتر	۴۰-۱۲۶	۳
۳	آز طراحی دیجیتال	۴۰-۲۰۳	۱
۴	آز مدارهای منطقی	۴۰-۲۰۶	۱
۵	مدارهای منطقی	۴۰-۲۱۲	۳
۶	طراحی سیستم های دیجیتال	۴۰-۲۲۳	۳
۷	معماری کامپیوتر	۴۰-۳۲۳	۳
۱۵	جمع واحدهای درسی که باید اخذ شود		

۶- سیستم های عامل

ردیف	نام درس	شماره درس	واحد
۱	از سیستم عامل	۴۰-۴۰۸	۱
۲	سیستم عامل	۴۰-۴۲۴	۳
۴	جمع واحدهای درسی که باید اخذ شود		

۷- رایانش شبکه-محور

ردیف	نام درس	شماره درس	واحد
۱	آز شبکه و انتقال داده	۴۰-۴۱۶	۱
۲	شبکه های کامپیوتری	۴۰-۴۴۳	۳
۴	جمع واحدهای درسی که باید اخذ شود		

۸- زبان های برنامه سازی

ردیف	نام درس	شماره درس	واحد
۱	طراحی زبانهای برنامه سازی	۴۰-۳۶۴	۳
۲	کامپایلر	۴۰-۴۱۴	۳
۶	جمع واحدهای درسی که باید اخذ شود		

۹- سیستم های هوشمند

ردیف	نام درس	شماره درس	واحد
۱	هوش مصنوعی	۴۰-۴۱۷	۳
۳	جمع واحدهای درسی که باید اخذ شود		

۱۰ مدیریت اطلاعات

واحد	شماره درس	نام درس	ردیف
۳	۴۰-۳۲۴	بازیابی پیشرفته اطلاعات	۱
۳	۴۰-۳۸۴	پایگاه داده ها	۲
۶	جمع واحدهای درسی که باید اخذ شود		

۱۱ مهندسی نرم افزار

واحد	شماره درس	نام درس	ردیف
۱	۴۰-۴۰۴	آز مهندسی نرم افزار	۵
۳	۴۰-۴۱۸	تحلیل و طراحی سیستمها	۶
۳	۴۰-۴۷۴	مهندسی نرم افزار	۷
۳	۴۰-۴۸۴	طراحی سیستم های شیء گرا	۸
۱۰	جمع واحدهای درسی که باید اخذ شود		

۱۲ کارآموزی و پروژه

واحد	شماره درس	نام درس	ردیف
۰	۴۰-۴۱۰	کارآموزی نرم افزار	۱
۳	۴۰-۷۱۰	پروژه نرم افزار	۲
۳	جمع واحدهای درسی که باید اخذ شود		

فهرست دروس اختیاری رشته مهندسی کامپیوتر - نرم افزار (مجموعاً ۱۲ واحد باید گذرانده شود)

ردیف	نام درس	شماره درس	واحد
۱	سیگنال ها و سیستم ها	۴۰-۲۴۲	۳
۲	مفاهیم پیشرفته در مهندسی نرم افزار	۴۰-۳۲۷	۳
۳	آداب فناوری اطلاعات	۴۰-۳۴۷	۳
۴	برنامه نویسی وب	۴۰-۴۱۹	۳
۵	سیستم های خیره	۴۰-۴۳۷	۳
۶	امنیت شبکه	۴۰-۴۴۲	۳
۷	گرافیک کامپیوتری	۴۰-۴۴۷	۳
۸	سیستم های اطلاعات مدیریت	۴۰-۴۶۴	۳
۹	مهندسی فناوری اطلاعات	۴۰-۴۶۷	۳
۱۰	شبیه سازی کامپیوتری	۴۰-۶۳۴	۳
۱۱	یکی از دروس کارشناسی ارشد دانشکده		۳
۱۲	یک درس از گرایش های دیگر دانشکده و یا دانشکده های دیگر		۳
جمع واحدهای دروس اختیاری که باید اخذ شود (از لیست فوق)			۱۲

ریز مواد دروس

دوره کارشناسی مهندسی کامپیوتر - نرم افزار

* مخصوص ورودی‌های ۱۳۸۸ به بعد *

{ این مجموعه تنها شامل دروس تخصصی ارائه شده توسط گروه نرم‌افزار می‌باشد }



کارگاه کامپیوتر (۴۰۱۰۸)

Computer Workshop

طراح درس: گروه نرم افزار*

مقطع: کارشناسی	گرایش: نرم افزار
نوع درس: اصلی	واحد: ۱
هم نیاز: -	پیش نیاز: -

کلیات

هدف این درس ارائه‌ی دانش و مهارت‌های لازم برای کاربری عمومی کامپیوتر برای دانشجویان رشته‌ی مهندسی کامپیوتر است که در ابتدای ورود به دوره باید گذرانده شود.

خلاصه مواد

- دانشجو باید مهارت‌های عمومی در کار با کامپیوتر شامل شناخت سخت‌افزار، سیستم عامل و نرم‌افزارهای متداول، برنامه‌نویسی تحت وب، اصول کلی شبکه و اینترنت و نحوه‌ی استفاده از آن‌ها را بیاموزد. عمده‌ی مفاهیم این درس شامل موارد زیر است:
۱. فهم کلی جایگاه سیستم عامل در کامپیوتر و آشنایی عمومی با سیستم عامل ویندوز (عملکردهای اولیه مانند کپی، جستجو در فایل‌ها، کلیک راست، کلیدها و میان‌برهای متداول، منوی start و امکانات موجود در آن، و ...)
 ۲. معرفی و استفاده از اینترنت و نامه الکترونیک
 ۳. اصول سخت‌افزار کامپیوتر (ساختار کلی و عملکرد هر جزء شامل ماوس، صفحه کلید، نمایشگر، کیس، برد اصلی، حافظه‌ها، پردازنده، سیستم خنک‌کننده، منبع تغذیه، نحوه‌ی ارتباطات اجزا، و ...)
 ۴. کار با نرم‌افزار Microsoft Word (ساخت یک سند جدید، قالب‌دهی به کلمه، پاراگراف، صفحه، آشنایی با style، افزودن جدول/تصویر/فرمول به متن، ویژگی‌های خاص برای نگارش فارسی، و ...)
 ۵. کار با نرم‌افزارهای Microsoft PowerPoint و Microsoft Excel (کاربرد PowerPoint و Excel، ایجاد و ویرایش فایل‌ها، مفهوم Slide Master در PowerPoint، پویانمایی‌ها در PowerPoint، قالب‌دهی سلول‌های صفحه در Excel، استفاده از فرمول‌ها و توابع در Excel، و ...)
 ۶. آشنایی عمومی با شبکه و اینترنت (ساختار کلی شبکه و اینترنت، موتورهای جستجو، نامه الکترونیک، مفاهیم اولیه مانند IP و URL، آشنایی با پروتکل‌های متداول مانند HTTP، FTP، IMAP، و آشنایی با دستورات پرکاربرد در شبکه مانند ping، ipconfig و ...)
 ۷. اصول کلی برنامه‌نویسی HTML (برنامه‌نویسی مقدماتی HTML، آشنایی با CSS، JavaScript، ویرایشگرهای HTML مانند FrontPage و ...)
 ۸. اصول کلی برنامه‌نویسی PHP (مفهوم برنامه‌نویسی server side، متغیرها در PHP، کار کردن با انواع داده‌ها در PHP، آشنایی با ساختارهای شرطی و حلقه در PHP، و ...)
 ۹. آشنایی با سیستم عامل لینوکس (مفهوم shell، جستجو در فایل‌ها، مجوزها، Help، فشرده‌سازی، دستورات مهم مانند cd، ls، cp، passwd، kill، chmode، su، grep و ...)
 ۱۰. آشنایی تکمیلی با سیستم عامل ویندوز (فرمان‌های مهم و سویچ‌ها در خط فرمان، متغیرهای محیطی، نوشتن فایل‌های BAT، Active Directory، Domain، Windows Registry، نرم‌افزارهای پرکاربرد مانند آنتی‌ویروس، Download Manager، محیط‌های برنامه‌نویسی، ابزارهای گرافیکی، و ...)

* ریز مواد این درس توسط آقای مهندس محمد علی هادوی تنظیم شده و توسط کمیته بازنگری دروس برنامه‌سازی (منتخب گروه نرم‌افزار) مورد پالایش و تکمیل قرار گرفته است.

مطالب قابل ارائه در این درس در قالب ۱۰ جلسه و یک تمرین پیشنهادی برای هر جلسه در ادامه آورده شده است.

۱ - اصول و مفاهیم کامپیوتر و سیستم عامل ویندوز

- ورود و خروج به/از سیستم
 - ویژگی‌های عمومی و اولیه‌ی ویندوز
 - ویژگی‌ها مختلف مانند کلیک راست، منوی Start، Taskbar، My computer، Desktop، و ...
 - جستجو در فایل‌ها و پوشه‌ها
 - عملیات متداول مانند copy، paste، cut و ...
 - میان‌برها و کلیدهای متداول مانند Ctrl+c، Ctrl+v، Ctrl+x، Ctrl+d، Win + e، Win + r، Win + Tab، Alt + Tab، Esc، PgUp، PgDn، Home، Alt+F4، Num Lock، Caps Lock، End، و ...
 - کاربردهای اولیه در منوی START مانند paint، calculator، command prompt، notepad، magnifier و ...
 - ایجاد، ذخیره‌سازی و بازکردن فایل‌ها (متنی / تصویری)
 - استفاده از اینترنت و کاربرد نامه الکترونیک شامل آشنایی اولیه با مرورگرها، وب‌گاه دانشگده، و وب‌گاه درس
- تمرین پیشنهادی:** دانشجویان باید در mailing list درس کارگاه کامپیوتر عضو شده و یک نامه‌ی الکترونیک برای آن شامل اطلاعات دانشجویی خود ارسال کنند.

۲ - آشنایی با اصول سخت‌افزار کامپیوتر

- ساختار کلی کامپیوتر و ارتباطات کلی اجزا شامل ماوس، صفحه کلید، کیس، و صفحه‌ی نمایش
 - ساختار درونی اجزای اصلی مانند
 - چگونگی نمایش اشیا در صفحه نمایش (نمایشگرهای LED، LCD و CRT)
 - عملکرد صفحه کلید و مکانیزم بافرینگ
 - عملکرد ماوس (مکانیکی و نوری)
 - نگاهی به محتوای Case و تشریح عملکرد منبع تغذیه، برد اصلی، پردازنده، سیستم خنک کننده، دیسک سخت، حافظه‌ی اصلی، کارت شبکه، کارت گرافیک، نحوه‌ی اتصال اجزا به برد اصلی
 - حافظه‌های جانبی مانند Flash memory، CD، DVD و ...
- تمرین پیشنهادی:** دانشجویان در گروه‌های دو یا سه نفره در یک فعالیت گروهی تحقیقی، در مورد یکی از اجزای سخت افزاری، اطلاعات کامل‌تری یافته و گزارشی مختصر (یک تا دو صفحه) تهیه می‌نمایند. (در صورت لزوم روش جستجو در اینترنت و منابع مفید معرفی گردد.)

۳ - MS Office - قسمت اول: MS Word

- آشنایی با منوها، ابزارها و میان‌برهای متداول در MS Word
- نگارش/ویرایش متن در MS Word
 - آشنایی با قلم‌ها و قالب‌های مختلف حروف مانند ایتالیک، زیرخط، رنگ متن، و ...
 - قالب‌دهی به پاراگراف‌ها (جهت متن، تنظیم فواصل از دو طرف، فاصله‌ی خطوط و ...)
 - افزودن اشیا اصلی مانند جدول و تصویر
 - بولت‌ها و شماره‌گذاری
- قالب‌دهی به صفحات شامل حاشیه‌های صفحه، جهت متن در صفحه، اندازه صفحه، تعیین حدود صفحه (page borders)، شماره صفحه، سرابند (header) و پانویس (footer) صفحه، Page Break، Section Break، Column Break، و ...
- کار کردن با سبک (style)
 - هدف سبک

- سبک‌های پاراگراف و کاراکترها
- کاربرد format painter
- میان‌برهای Ctrl+Q و Ctrl + Space
- ایجاد فهرست مطالب
- درج فرمول
- جنبه‌های ویژه‌ی فارسی‌نویسی مانند تغییر زبان نگارش با Alt+Shift و Ctrl +. برای درج فاصله مجازی (optional hyphen)
- تبدیل سند MS Word به pdf

تمرین پیشنهادی: دانشجویان باید یک سند حداقل دو صفحه‌ای که ترکیبی از نگارش فارسی و انگلیسی است را حروف‌چینی کرده و مطالب آموخته شده در این جلسه را در نگارش بکار گیرند. سرعت نوشتن نیز از موارد قابل طرح در هدف این تمرین است به گونه‌ای که برای آزمون عملی میان‌ترم سرعت ۱۵-۲۰ کلمه در دقیقه مدنظر است. این نرخ باید تا آزمون عملی پایان ترم به حدود ۲۵-۳۰ کلمه در دقیقه افزایش یابد.

۴- MS Office – قسمت دوم: MS PowerPoint & Excel

- Microsoft PowerPoint
 - کاربرد PowerPoint
 - تولید یک ارائه‌ی (presentation) جدید
 - طرح‌ها و رنگ‌ها در ارائه
 - پویانمایی در ارائه
 - کاربرد Slide Master
- Microsoft Excel
 - Execl چیست؟
 - کاربردهای Excel
 - توصیف ساختار داده‌ای جدولی و درج داده در سلول‌های جداول
 - قالب‌بندی سلول‌های جدول
 - استفاده از فرمول‌ها (عملگرها و توابع ریاضی، و توابع تجمعی و ...)
 - تولید نمودار در Excel

تمرین پیشنهادی: دانشجویان باید بتوانند یک ارائه PowerPoint تولید کرده و تا حد امکان از نکات آموخته شده در تولید آن استفاده کنند. همچنین دانشجویان باید یک صفحه Excel (worksheet) تولید کرده و در آن، ضمن استفاده از برخی توابع ریاضی، نمودارهایی مبتنی بر داده‌های موجود تولید نمایند.

۵- شبکه و اینترنت

- شبکه چیست و چرا از شبکه استفاده می‌کنیم؟
- اینترنت چیست؟
- استفاده از اینترنت (وب، مرورگرها، موتورهای جستجو و نامه الکترونیک)
- آشنایی با مفاهیم URL، IP و MAC address
- مؤلفه‌های اصلی در شبکه‌ها (سیستم‌های سرویس‌دهنده و سرویس‌گیرنده، هاب، سوئیچ، مسیریاب، کابل و ...)
- پروتکل‌ها (مفهوم پروتکل و پروتکل‌های اصلی مانند HTTP، FTP، SMTP، POP3، IMAP، HTTPS، SFTP، SMTPS، POP3S، IMAPS)
- استفاده از دستورات متداول در شبکه مانند ping، nslookup، tracert، ipconfig و ...
- استفاده از سرویس whois
- شناخت ساختار شبکه دانشکده و دانشگاه
- سرویس‌های DNS و DHCP

تمرین پیشنهادی: به هر دانشجو یک مفهوم/پروتکل/سرویس در شبکه معرفی شود تا دانشجو ضمن تحقیق در آن مورد، گزارشی یک صفحه‌ای فارسی تهیه نموده و به صورت فایل pdf ارسال نماید.

۶- برنامه‌نویسی مقدماتی وب

- برنامه‌نویسی HTML
 - HTML و کاربردهای آن
 - HTML در Tag ها
 - ساختار کلی HTML
 - شناخت Tag های اصلی مانند title, ol, br, a, img, li, ul, table, b, i, u, font, p, pre, xmp, pre, meta و کاراکترهای خاص مانند
- اصول CSS
 - Style Sheet چیست؟
 - مقدمه‌ای کوتاه بر CSS
 - Tag <link> در HTML (external css)
- برنامه‌نویسی جاوا اسکریپت
 - توصیف (Document Object Model) DOM
 - رویدادها (events) و پیاده‌سازی توسط جاوا اسکریپت
 - نوشتن برنامه‌ای کوتاه با جاوا اسکریپت
- ویرایش‌گرهای HTML مانند FrontPage

تمرین پیشنهادی: هر دانشجو موظف با ساخت یک homepage ساده برای خودش است (نیازی به بارگذاری آن در سایت CE نیست). این homepage باید شامل حداقل سه صفحه که به هم پیوند شده‌اند و همچنین شامل CSS و جاوااسکریپت باشد. دانشجویان نباید از ویرایش‌گرهای WYSIWYG یا کدهای از پیش موجود جاوا اسکریپت استفاده کنند. پیشنهاد می‌شود که این homepage با محتوایی به عنوان رزومه دانشجویان تهیه گردد.

۷- سیستم عامل لینوکس (قسمت اول: مقدمات)

- SSH و استفاده از آن برای گرفتن Linux Shell
- Shell چیست (جایگاه آن در برابر kernel)؟
- معرفی Shell های مشهور مانند C, Z, Bourne, BASH, KORN و TC
- استفاده از Tab و کلیدهای جهت‌نما
- ویژگی فایل در لینوکس (همه چیز در لینوکس فایل است، case sensitive بودن فایل‌ها در لینوکس)
- دستورات اصلی در Shell
 - دستورات مرتبط با فایل و دایرکتوری (همراه با سوئیچ‌ها) مانند cd, ls, mv, cp, rm, mkdir, pwd, ln, touch, cat, vim, head, tail
 - معنای wildcard در لینوکس (*, ?, و [])
 - دستورات مرتبط با جستجو در فایل‌ها مانند find, locate, updatedb, whereis, و which
 - دستورات مرتبط با مجوزها (chmod, chgrp, chown)
 - دستورات مرتبط با امنیت (su, passwd, logout)
 - دستورات مرتبط با منابع سیستم (ps, kill, free, df, du)
 - سایر دستورات متداول مانند echo, who, whoami, finger, write, talk, mesg, gcc
- تولید یک پوشه‌ی "public_html" و اعطای مجوزها به آن و بارگذاری فایل‌های HTML در آن
- تکنیک‌های متداول در لینوکس
 - معنای ~ در لینوکس

- Redirection (>, <, >>) و piping (|) با استفاده از sort و less
- اجرای چند دستور با استفاده از semicolon (;)، && و ||
- اجرای دستورات در پیش‌زمینه (background) با استفاده از &
- استفاده از backtick (`) و (\$) برای ارسال خروجی یک دستور به دستور بعدی
- استفاده از backslash (\) برای دستورات بیشتر از یک خط
- Help در لینوکس با دستورات man, info, how-to, --help, whatis, و apropos

تمرین پیشنهادی: هر دانشجو باید یک دستور جدید در لینوکس و عملکرد آن را توضیح دهد و نتیجه را در قالب یک گزارش فارسی به صورت pdf (یک صفحه) ارائه دهد.

۸- سیستم عامل لینوکس (قسمت دوم: مباحث تکمیلی)

- آلیاس‌ها (aliases) و نحوه‌ی استفاده از آنها (فایل‌های .bashrc) و استفاده از دستور reset
- مشاهده‌ی تاریخچه‌ی دستورات اجرا شده و دستور history
- عبارات منظم در لینوکس و کار کردن با دستورات grep, awk, sed, و xargs
- فشرده‌سازی فایل‌ها و فهرست‌ها و باز کردن فایل‌ها و فهرست‌های فشرده شده با gzip, bzip, و tar
- کاربرد نویسه‌های bash (\$, if, for, while, [], case)

تمرین پیشنهادی: دانشجویان در گروه‌های دو یا سه نفری باید یک نویسه‌ی bash بنویسند. این نویسه باید شامل دستورات متعدد باشد.

۹- برنامه‌نویسی PHP

- مفهوم برنامه‌نویسی server side و جایگاه PHP
- فرم‌های HTML
- توصیف جریان متقابل اطلاعات بین صفحات وب و برنامه‌ی PHP
- تعریف متغیرها در PHP و استفاده از دستور echo
- کار کردن با اعداد، رشته‌ها، تاریخ، زمان و آرایه‌ها در PHP
- عبارات شرطی و حلقه در PHP
- نصب PHP به همراه یک کارگزار وب (مانند Apache) برای اجرای server side یک برنامه‌ی PHP

تمرین پیشنهادی: هر دانشجو باید به homepage ساخته شده‌ی خود (در جلسه‌ی ششم)، یک کد PHP اضافه نموده و آن‌را در کارگزار CE بارگذاری نماید.

۱۰- مباحث تکمیلی در ویندوز

- خط فرمان ویندوز
 - دستورات cd, dir, rd, del, rmdir, tree, type, edit, md و سوئیچ‌های مهم دستورات
 - متغیرهای محیطی و دستور SET
 - نوشتن فایل‌های BAT (افزودن FOR و IF به فایل‌های BAT)
- رجیستری ویندوز (توصیف عناصر مهم مانند HKCU, HKLM و ...)
- Active Directory و Domain (از جزئیات فراوان صرف نظر شود)
- نرم‌افزارهای متداول روی ویندوز مانند ضدبدافزارها، ابزارهای mount کردن CD/DVD، برنامه‌های گرافیکی، Download Managers، محیط‌های برنامه‌نویسی و ...

تمرین پیشنهادی: هر دانشجو باید یک فایل BAT با یک کارکرد مشخص (مثلاً تغییر مقادیری در رجیستری) بنویسد. ارائه‌ی گزارشی فارسی که نتیجه‌ی تحقیق روی جنبه‌های Active Directory است نیز می‌تواند به عنوان یک تمرین با رویکرد تحقیقاتی مطرح گردد.

آزمون - تمرین

پیشنهاد می‌شود نمره‌ی نهایی بر اساس آزمون تئوری در پایان ترم (و میان‌ترم) (۵۰٪)، ۱۰ تمرین هفتگی (۳۵٪) و آزمون‌های عملی (۱۵٪) تعیین گردد. با توجه به ماهیت عملی درس، تعیین تمرین‌های هفتگی از مطالب تدریس شده و اختصاص بخشی از نمره‌ی نهایی بر اساس تمرین‌های هفتگی و همچنین دو آزمون عملی بر اساس مطالب تدریس شده (به‌ویژه در حوزه‌ی نگارش و حروف‌چینی صحیح) توصیه می‌گردد.

مراجع اصلی

- C. Newman, *SAMS Teach Yourself PHP in 10 Minutes*, Sams Publishing, 2005. (Lessons 1-12, 16, and Appendix A)
- D. Hayes, *Sams Teach Yourself HTML in 10 Minutes*, 4th Edition, Sams publishing, 2006. (Lessons 1-8, 13-15)
- R. Weakley, *Sams Teach Yourself CSS in 10 Minutes* Sams Publishing, 2005. (for additional information)
- B. Forta, *Sams Teach Yourself Regular Expressions in 10 Minutes*, Sams Publishing, 2004. (for additional information)
- R. Shimonski, *SAMS Teach Yourself Unix in 10 Minutes*, Sams Publishing, 2005. (Lessons 1, 2, 4-8, 10-16, 18, 19)
- J. Andrews, *A+ Guide to Managing & Maintaining Your PC*, 7th Edition, Course Technology, 2009.
- *IT Essentials PC Hardware and Software Course Booklet, Version 4.1*, 2nd Edition, Cisco Press.



مبانی برنامه‌سازی (C) (۴۰۱۵۳) Fundamentals of Programming (C)

طراح درس: گروه نرم‌افزار*

مقطع: کارشناسی	گرایش: نرم‌افزار
نوع درس: اصلی	واحد: ۳
هم‌نیاز: -	پیش‌نیاز: -

کلیات

هدف از این درس، آشنایی دانش‌جویان با برنامه‌نویسی به زبان C، نوشتن کد مهندسی‌ساز (برنامه‌سازی ساختارمند، توضیح‌گذاری، جدانویسی)، و توانایی پیاده‌سازی شبه‌کد است.

ریز مواد

خلاصه: مفاهیم اولیه، تاریخچه‌ی کامپیوتر، معرفی اجزای اصلی کامپیوتر (سخت‌افزار - نرم‌افزار)، سیستم‌های عددی در کامپیوتر، نمایش داده‌های عددی و غیر عددی، آشنایی با الگوریتم، آشنایی با برنامه‌سازی ساخت‌یافته به زبان C، ثابت‌ها، متغیرها، عبارات‌های محاسباتی و منطقی، انواع دستورات کنترلی (دستورات انتخاب، دستورات تکرار)، تبدیلات داده‌ها، بردارها، ماتریس‌ها، الگوریتم‌های متداول مانند روش‌های جست‌وجو و مرتب‌سازی، برنامه‌های فرعی (توابع، سربارگذاری توابع)، آشنایی با آزمون و خطایابی برنامه‌ها، اشاره‌گرها، دستورات ورودی و خروجی (پرونده‌ها)، آشنایی با رده‌ها و انتزاع داده.

۱. مفاهیم اولیه (۱ جلسه)

- تاریخچه رشد کامپیوتر (سیستم‌های عامل، زبان‌های برنامه‌سازی)
- معرفی اجزای اصلی کامپیوتر
- تاریخچه C/C++
- برنامه‌سازی ساخت‌یافته و نوشتن کد مهندسی‌ساز
- ساختار و مراحل ساخت و اجرای برنامه

۲. محاسبات در کامپیوتر (۱ جلسه)

- سیستم‌های عددی (دودویی، ده‌دهی، مبنای شانزده)
- جمع و تفریق در مبنایها
- تبدیل مبنایها

۳. مقدمات برنامه‌سازی (۱ جلسه)

- متغیرها، انواع داده
- دستورات ورودی/خروجی
- آشنایی با کدنویسی خوب (توضیح‌گذاری، جدانویسی)
- محاسبات ریاضی (جمع، تفریق، ضرب، تقسیم، باقی‌مانده)

* این سیلابس توسط آقای مهندس شروین دانش‌پژوه تنظیم شده و توسط کمیته بازنگری دروس برنامه‌سازی (منتخب گروه نرم‌افزار) مورد پالایش و تکمیل قرار گرفته است.

- اولویت محاسبات
 - مقایسه
۴. فرمت‌بندی ورودی/خروجی (۱ جلسه)
- قالب‌بندی خروجی با `printf`
 - کنترل دقت نمایش اعداد در خروجی
 - چاپ اعداد صحیح، اعداد ممیزدار، رشته‌ها، و کاراکترها
 - خواندن قالب‌بندی شده از ورودی با `scanf`
۵. آشنایی با الگوریتم، فلوچارت و شبه کد (۲ جلسه)
۶. دستورات (۳ جلسه)
- دستورات کنترلی
 - دستورات انتخاب (`if, if/else, switch`)
 - دستورات تکرار (`while, for, do/while`)
 - `continue, break`
 - عملگر انتخاب: `?:`
 - تبدیل داده‌ها
 - ثابت‌ها (`const`)
 - عملگرهای منطقی
 - استفاده از `typename`
۷. توابع (۴ جلسه)
- توابع ریاضی
 - توابع تصادفی
 - توابع زمان
 - الگوی توابع (`prototype`)
 - رده‌های ذخیره‌سازی (`static, register, extern`)
 - حوزه‌های تعریف (`file, function, block`)
 - پیشته در فراخوانی تابع
 - توابع بازگشتی
 - مقایسه توابع بازگشتی و غیربازگشتی
 - پرونده‌های `header`
۸. آشنایی با آزمون و خطایابی برنامه (۱ جلسه)
- استفاده از دستورات خروجی و `__LINE__` و `__FILE__` برای پیدا کردن خطا
 - استفاده از امکانات IDE برای پیدا کردن خطا
 - `debug window, next step, step into, step over, run to cursor`
۹. آرایه‌ها (۵ جلسه)
- تعریف و به‌کارگیری آرایه‌ها
 - ارسال آرایه به تابع
 - جستجوی دودویی در آرایه
 - مرتب‌سازی آرایه
 - آرایه‌های چندبعدی
 - تعریف ثابت‌ها با استفاده از `define`
۱۰. اشاره‌گرها (۳ جلسه)

- عملگرهای اشاره‌گری
- فراخوانی با ارجاع توسط اشاره‌گرها
- استفاده از const در اشاره‌گرها
- عملگر sizeof
- محاسبات آدرس بر روی اشاره‌گرها (جمع، تفریق)
- ارتباط بین اشاره‌گرها و آرایه‌ها
- آرایه‌ای از اشاره‌گرها

۱۱. کاراکترها و رشته‌ها (۱ جلسه)

- تبدیل کاراکترها
- توابع رشته‌ای
- دست‌کاری رشته‌ها

۱۲. ساختارها (۲ جلسه)

- تعریف ساختار (struct)
- دسترسی به اعضای ساختار
- ارسال ساختارها به توابع
- ایجاد نام‌های مترادف با typedef
- نوع داده‌ی union
- ثابت‌های شمارشی (enum)
- عملگرهای بیتی

۱۳. آشنایی با زبان C++ (۱ جلسه)

- ورودی و خروجی با استفاده از جویبارها
- فضای نام‌ها، استفاده از using
- عملگر تغییر حوزه "::~"
- انواع فراخوانی توابع
 - فراخوانی با مقدار
 - فراخوانی با ارجاع
- توابع درون‌خط (inline)
- تعریف مقدار پیش فرض در توابع
- سربارگذاری توابع

۱۴. آشنایی با رده‌ها و انتزاع داده (۳ جلسه)

- مفهوم قابلیت استفاده‌ی مجدد در نرم‌افزار
- رده‌ها (class)
- اعضای داده‌ای و توابع عضو
- حوزه‌های public و private
- عملگرهای دسترسی به اعضای رده
- سازنده‌ها
- مقداردهی اشیاء
- ارسال/دریافت اشیاء به/از توابع
- رده‌های نمونه (اختیاری)

- آشنایی با string از کتاب‌خانه‌ی استاندارد
- آشنایی با vector از کتاب‌خانه‌ی قالب استاندارد

- مباحث تکمیلی (اختیاری)
 - سربارگذاری عملگرها
 - استفاده از قالبها

۱۵. پروندهها (۱ جلسه)

- خواندن و نوشتن در پروندههای ترتیبی
- جویبارهای ورودی و خروجی
- خواندن و نوشتن در پروندههای تصادفی (اختیاری)

۱۶. مباحث اختیاری (۱ جلسه - در صورت فرصت)

- آشنایی با فرایند تولید نرم افزار
- پیش پردازش گرها در C

آزمون - تمرین

تمرین ۱. محاسبات در کامپیوتر، مقدمات برنامه سازی، قالب بندی ورودی/خروجی

تمرین ۲. الگوریتم و فلوچارت، دستورات کنترلی

تمرین ۳. توابع و آرایه ها

تمرین ۴. اشاره گرها، کاراکترها و رشته ها

تمرین ۵. رده ها و انتزاع داده، پرونده ها

عنوان	توضیحات	نمره از ۲۰
تمرین	دست کم ۴ تمرین	۴
آزمونک	دست کم ۴ آزمونک	۲
پروژه	پروژه در دو مرحله در طول نیم سال به دانش جویان ارائه می شود.	۳
آزمون میان ترم		۴
آزمون پایانی		۷
فعالیت کلاسی	مشارکت و فعالیت موثر در کلاس های درس و حل تمرین	(+۱)

مرجع اصلی

- P. Deitel, H. Deitel, C: *How to Program*, 6th Edition, Prentice Hall, 2009.



مقطع: کارشناسی	گرایش: نرم‌افزار
نوع درس: اصلی	واحد: ۳
هم‌نیاز: -	پیش‌نیاز: مبانی برنامه‌سازی

کلیات

هدف از این درس، آشنایی دانش‌جویان با مفاهیم، ساختارها، و تکنیک‌هایی از ریاضیات گسسته است که به‌طور گسترده در علوم و مهندسی کامپیوتر مورد استفاده قرار می‌گیرند. ایجاد مهارت‌های زیربنایی از جمله فهم و ساخت اثبات‌های دقیق ریاضی، تفکر خلاقانه در حل مسائل، آشنایی با نتایج اولیه در نظریه‌ی اعداد، منطق، ترکیبیات، و نظریه‌ی گراف‌ها، و نیز فراهم آوردن پیش‌نیاز ریاضی موردنیاز برای بسیاری دیگر از دروس ارائه‌شده در گرایش‌های مختلف مهندسی کامپیوتر از اهداف این درس به‌شمار می‌رود.

ریز مواد

۱. منطق (۳ جلسه)
 - اصول اولیه‌ی منطق، گزاره‌ها، گزاره‌های هم‌ارز
 - گزاره‌نماها، سورها، اصول استنتاج
 - روش‌های اثبات
۲. نظریه‌ی توابع و مجموعه‌ها (۲ جلسه)
 - مبانی نظریه‌ی مجموعه‌ها، عملگرهای مجموعه‌ای، مجموعه‌های شمارا و ناشمارا
 - توابع یک‌به‌یک و پوشا، ترکیب توابع، معکوس توابع، دنباله‌ها
۳. نظریه‌ی اعداد (۳ جلسه)
 - همنهشتی، بزرگ‌ترین مقسوم‌علیه مشترک، اعداد اول
 - الگوریتم‌های حسابی، الگوریتم اقلیدسی
 - مقدمه‌ای بر نظریه‌ی رمزنگاری
۴. استقرا (۲ جلسه)
 - استقرای ریاضی، اصل خوش‌ترتیبی
 - استقرای قوی، استقرای ساختاری
۵. شمارش (۵ جلسه)
 - اصول اولیه‌ی شمارش، جای‌گشت و ترکیب
 - ضرایب دوجمله‌ای، جای‌گشت‌ها و ترکیب‌های با تکرار
 - اصل طرد و شمول
 - اصل لانه‌ی کبوتری
 - (اختیاری) روش‌های تولید جای‌گشت و ترکیب
۶. روابط بازگشتی (۳ جلسه)
 - مسائل بازگشتی
 - حل روابط بازگشتی (همگن و غیر همگن)

- توابع مولد
- ۷. رابطه‌ها (۲ جلسه)
 - رابطه‌ها و خواص آن‌ها، نمایش رابطه‌ها، ترکیب روابط
 - رابطه‌های هم‌ارزی، بستارها
- ۸. ترتیب جزئی (۱ جلسه)
 - مجموعه‌های با ترتیب جزئی، نمودار هاس، شبکه‌ها، مرتب‌سازی توپولوژیکی
- ۹. جبر بول (۱ جلسه)
 - خواص جبر بول، توابع بولی، نمایش توابع بولی
- ۱۰. گراف‌ها (۳ جلسه)
 - تعاریف اولیه، گراف‌های خاص، گراف‌های دوبخشی، نمایش گراف‌ها، یک‌ریختی گراف‌ها
 - مسیرها و همبندی، مسیرهای اویلری و همیلتنی
 - گراف‌های مسطح، قضیه‌ی اویلر، رنگ‌آمیزی گراف‌ها
- ۱۱. درخت‌ها (۱ جلسه)
 - درخت‌ها و جنگل‌ها، درخت‌های خاص، درخت‌های ریشه‌دار، درخت‌های پوشا
- ۱۲. ساختارهای جبری (۱ جلسه)
 - تکواره‌ها، حلقه‌ها، گروه‌ها، گروه‌های آبدلی
- ۱۳. مدل‌سازی محاسبات (۳ جلسه)
 - زبان‌ها و گرامرها، ماشین‌های با حالات متناهی
 - تشخیص زبان‌ها، زبان‌های منظم
 - (اختیاری) ماشین تورینگ

آزمون - تمرین

- تمرین نظری: ۱۵٪ نمره
- آزمون‌ها (میان‌ترم، پایان‌ترم، و آزمونک‌ها): ۸۵٪ نمره

مرجع اصلی

- K. H. Rosen, *Discrete Mathematics and Its Applications*, 6th Edition, McGraw-Hill, 2007.

مرجع کمکی

- R. P. Grimaldi, *Discrete and Combinatorial Mathematics: An Applied Introduction*, 5th Edition, Addison-Wesley, 2004.



محاسبات عددی (۴۰۲۱۵)

Numerical Computations

طراح درس: گروه نرم‌افزار*

مقطع: کارشناسی	گرایش: نرم‌افزار
نوع درس: اصلی	واحد: ۳
هم‌نیاز: -	پیش‌نیاز: معادلات دیفرانسیل

کلیات

هدف این درس، آشنا کردن دانشجویان کارشناسی نرم‌افزار با مفاهیم خطاها، دلایل، انواع و امکان انتشار آن‌ها است. همچنین این درس معرف روش‌های عددی است که برای تخمین مسائل علمی رشته‌های مختلف علوم و مهندسی به کار می‌روند و محاسبه دقیق آن‌ها با روش‌های معمول ریاضی مقدور نمی‌باشد. این درس به طرح، تحلیل و مقایسه این روش‌ها از نظر میزان دقت، درجه همگرایی و سایر موارد می‌پردازد. استفاده از محیط‌های نرم‌افزاری کارآمد برای حل مسائل، مقایسه روش‌های عددی و نمایش گرافیکی نتایج برای خلاصه‌سازی و جمع‌بندی آن‌ها از اهداف دیگر این درس است.

با توجه به اینکه در حال حاضر بالابردن دقت محاسبات در بسیاری از مسائل مربوط به مهندسی کامپیوتر از اهمیت بالایی برخوردار است، ساختار درس بر معرفی روش‌های مختلف عددی و مقایسه دقت و سرعت آنها تمرکز دارد.

ریز مواد

- آشنایی با محیط نرم‌افزاری MATLAB: محاسبات مربوط به ماتریس‌ها- بردارها و ترسیم- M فایلها- تعریف توابع- آشنایی با برخی توابع MATLAB (۲ جلسه)
- خطاها: سیستم ممیز شناور- منشأ خطاها- خطاهای نسبی و مطلق- خطاهای گرد کردن، ذاتی و برشی- انتشارخطا- ناپایداری در محاسبات عددی (۲ جلسه)
- حل عددی معادلات غیرخطی: پیدا کردن ریشه‌های تابع تک متغیره غیرخطی- روش‌های عددی حل معادلات غیرخطی شامل: (روش تنصیف، نابه‌جایی، نیوتن_رافسون، تکرار ساده یا نقطه ثابت و روش وتری یا خط قاطع)- نرخ همگرایی روش‌های مختلف- شرط‌های لازم و کافی برای همگرایی روش‌های نیوتن_رافسون، روش وتری و روش تکرار ساده- روش هرزنر- روش تعمیم‌یافته نیوتن_رافسون برای حل دستگاه معادلات غیرخطی- اثبات شهودی و ریاضی روش‌های ذکر شده (۵ جلسه)
- درونیابی، برونیابی و برازش منحنی:
 - روش‌های مختلف درونیابی شامل: (لاگرانژ، تفاضلات تقسیم‌شده نیوتن، تفاضلات پیشرو، پسرو و مرکزی نیوتن)، تحلیل خطای محاسباتی در روش‌های ذکر شده- اثبات روش‌های گفته شده (۵ جلسه)
 - برازش منحنی توسط روش کمترین مربعات، برازش منحنی‌های مختلف با کمک خطی‌سازی (۱ جلسه)
 - برونیابی (۱ جلسه)
- انتگرال گیری و مشتق‌گیری عددی:
 - روش‌های مختلف انتگرال‌گیری عددی شامل: روش مستطیلی، دوزنقه، سیمپسون، گاوس_لژاندر، روش نقطه‌میان و روش رامبرگ - تحلیل میزان خطای روش‌های ذکر شده- اثبات ریاضی و شهودی برای روش‌های مذکور (۴ جلسه)
 - مشتق‌گیری عددی: ارائه روش‌های عددی مختلف برای مشتق‌گیری عددی و تحلیل مرتبه خطای آن‌ها، استفاده از برونیابی ریچاردلسون برای بهبود نتایج مشتق‌گیری عددی (۱ جلسه)
- حل معادلات دیفرانسیل معمولی:

* ریز مواد این درس توسط آقای مهندس محمد غریب تنظیم شده و توسط گروه نرم‌افزار مورد بازبینی و تصویب قرار گرفته است.

- روش‌های تک‌گامی مانند: (تایلور، اویلر، روش اصلاح‌شدهٔ اویلر و روش‌های رانگ_کوتا مرتبه ۲ (روش هیون، نقطه میانی و اصلاح شدهٔ اویلر) مرتبه ۳ و مرتبه ۴) - تبدیل معادلات دیفرانسیل درجات بالاتر به دستگاه معادلات دیفرانسیل خطی (۳ جلسه)
- روش‌های چندگامی مانند روش آدامز_مولتون - مقایسهٔ روش‌ها (۱ جلسه)
- ۷. حل عددی دستگاه معادلات خطی:
 - مقدمه‌ای بر ماتریس‌ها (۱ جلسه)
 - روش‌های مستقیم مانند: (روش کرامر، روش حذف گاوسی، تجزیهٔ مثلثی LU (روش‌های چولسکی، دولیتل و کروت)) (۴ جلسه)
 - روش‌های تکرار شونده مانند: (روش ژاکوبی، روش گاوس_سایدل) (۱ جلسه)
 - مقادیر ویژه و بردار ویژه - روش توانی - قضیهٔ گرچ_گورین (۱ جلسه)

نکته: جلسات کلاسهای سه واحدی یک ساعت و نیم می‌باشند.

آزمون - تمرین - پروژه

- آزمون: آزمون‌های میان ترم و پایان ترم (۵۰٪ کل نمره)
- تمرین: ۶ تمرین نظری (حل مسائل با روش‌های عددی)، ۶ تمرین عملی با محیط نرم‌افزاری MATLAB؛ تمرینات در طول نیمسال تحویل داده می‌شوند. (۴۰٪ کل نمره)
- پروژه: موضوع پروژه با کمک استاد انتخاب می‌شود. پروژه می‌تواند پژوهشی ویا عملی (با استفاده از نرم‌افزار MATLAB و یا نرم افزار مشابه) باشد. پس از انجام کار، نتیجه پروژه پژوهشی در قالب گزارش ارائه داده می‌شود. (۱۰٪ کل نمره)

مراجع اصلی

- S. Pal, *Numerical Methods Principles, Analysis and Algorithms*, ISBN: 978-0-19-569375-1, Oxford University Press, 2010.
- J. R. Hauser, *Numerical Methods for Nonlinear Engineering Models*, ISBN: 978-1-4020-9919-9, Springer, 2009.
- M. K. Jain, S. R. K. Iyengar, and R. K. Jain, *Numerical Methods for Scientific and Engineering Computation*, ISBN: 81-224-2001-X, 5th Edition, New Age International Publisher, 2007 .
- J. H. Mathews, K. D. Fink, *Numerical Methods Using Matlab*, 4th Edition, ISBN: 9780130652485, Pearson Prentice Hall, 2004.
- B. Hahn, D. Valentine, *Essential Matlab for Engineers and Scientists*, 4th Edition, Elsevier Science, 2009.



محاسبات عددی (۴۰۲۱۶)

Computations Numerical

طراح درس: گروه نرم افزار *

مقطع: کارشناسی	گرایش: سخت افزار - فناوری اطلاعات
نوع درس: اصلی	واحد: ۲
هم نیاز: -	پیش نیاز: معادلات دیفرانسیل

کلیات

هدف این درس، آشنا کردن دانشجویان کارشناسی سخت افزار و فناوری اطلاعات با مفاهیم خطاها، دلایل، انواع و امکان انتشار آن‌ها است. همچنین این درس معرف روش‌های عددی است که برای تخمین مسائل علمی رشته‌های مختلف علوم و مهندسی به کار می‌روند و محاسبه دقیق آن‌ها با روش‌های معمول ریاضی مقدور نمی‌باشد. این درس به طرح، تحلیل و مقایسه این روش‌ها از نظر میزان دقت، درجه همگرایی و سایر موارد می‌پردازد.

استفاده از محیط‌های نرم‌افزاری کارآمد برای حل مسائل، مقایسه روش‌های عددی و نمایش گرافیکی نتایج برای خلاصه‌سازی و جمع‌بندی آن‌ها از اهداف دیگر این درس است.

با توجه به اینکه در حال حاضر بالابردن دقت محاسبات در بسیاری از مسائل مربوط به مهندسی کامپیوتر از اهمیت بالایی برخوردار است، ساختار درس بر معرفی روش‌های مختلف عددی و مقایسه دقت و سرعت آنها تمرکز دارد.

ریز مواد

۱. آشنایی با محیط نرم‌افزاری MATLAB: محاسبات مربوط به ماتریس‌ها- بردارها و ترسیم- M فایلها- تعریف توابع- آشنایی با برخی توابع MATLAB (۲ جلسه)
۲. خطاها: سیستم ممیز شناور- منشأ خطاها- خطاهای نسبی و مطلق- خطاهای گرد کردن، ذاتی و برشی- انتشار خطا- ناپایداری در محاسبات عددی (۲ جلسه)
۳. حل عددی معادلات غیرخطی: پیدا کردن ریشه‌های تابع تک متغیره غیرخطی- روش‌های عددی حل معادلات غیرخطی شامل: (روش تنصیف، نابه‌جایی، نیوتن-رافسون، تکرار ساده یا نقطه ثابت و روش وتری یا خط قاطع)- نرخ همگرایی روش‌های مختلف- شرط‌های لازم و کافی برای همگرایی روش‌های نیوتن-رافسون، روش وتری و روش تکرار ساده- روش هرنز- روش تعمیم یافته نیوتن-رافسون برای حل دستگاه معادلات غیرخطی- اثبات شهودی روش‌های ذکر شده (۵ جلسه)
۴. درونیایی، برونایی و برازش منحنی:
 - روش‌های مختلف درونیایی شامل: (لاگرانژ، تفاضلات تقسیم شده نیوتن، تفاضلات پیشرو، پسرو و مرکزی نیوتن)، تحلیل خطای محاسباتی در روش‌های ذکر شده (۵ جلسه)
 - برازش منحنی توسط روش کمترین مربعات، برازش منحنی‌های مختلف با کمک خطی‌سازی (۱ جلسه)
 - برونایی (۱ جلسه)
۵. انتگرال گیری و مشتق گیری عددی:
 - روش‌های مختلف انتگرال گیری عددی شامل: روش مستطیلی، دوزنقه، سیمپسون، گاوس-لژاندر، روش نقطه‌میانی و روش رامبرگ - تحلیل میزان خطای روش‌های ذکر شده- اثبات شهودی برای روش‌های مذکور (۴ جلسه)
 - مشتق گیری عددی: ارائه روش‌های عددی مختلف برای مشتق گیری عددی و تحلیل مرتبه خطای آن‌ها، استفاده از برونایی ریچاردسون برای بهبود نتایج مشتق گیری عددی (۱ جلسه)

* ریز مواد این درس توسط آقای مهندس محمد غریب تنظیم شده و توسط گروه نرم‌افزار مورد بازبینی و تصویب قرار گرفته است.

۶. حل معادلات دیفرانسیل معمولی:

- روش‌های تک‌گامی مانند: (تایلور، اویلر، روش اصلاح‌شده اویلر و روش‌های رانگ_کوتا مرتبه ۲) روش هیون، نقطه میانی و اصلاح شده اویلر) مرتبه ۳ و مرتبه ۴- تبدیل معادلات دیفرانسیل درجات بالاتر به دستگاه معادلات دیفرانسیل خطی (۳ جلسه)
- روش‌های چندگامی مانند روش آدامز_مولتون- مقایسه روش‌ها (۱ جلسه)

۷. حل عددی دستگاه معادلات خطی:

- مقدمه‌ای بر ماتریس‌ها (۱ جلسه)
- روش‌های مستقیم مانند: (روش کرامر، روش حذف گاوسی، تجزیه مثلثی LU (روش‌های چولسکی، دولیتل و کروت)) (۴ جلسه)
- روش‌های تکرار شونده مانند: (روش ژاکوبی، روش گاوس_سایدل) (۱ جلسه)
- مقادیر ویژه و بردار ویژه- روش توانی- قضیه گرچ‌گورین (۱ جلسه)

نکته: جلسات کلاسهای دو واحدی یک ساعته می‌باشند.

آزمون - تمرین

- آزمون: آزمون‌های میان ترم و پایان ترم (۵۰٪ کل نمره)
- تمرین: ۶ تمرین نظری (حل مسائل با روش‌های عددی)، ۶ تمرین عملی با محیط نرم‌افزاری MATLAB؛ تمرینات در طول نیمسال تحویل داده می‌شوند. (۵۰٪ کل نمره)

مراجع اصلی

- S. Pal, *Numerical Methods Principles, Analysis and Algorithms*, ISBN: 978-0-19-569375-1, Oxford University Press, 2010.
- J. R. Hauser, *Numerical Methods for Nonlinear Engineering Models*, ISBN: 978-1-4020-9919-9, Springer, 2009.
- M. K. Jain, S. R. K. Iyengar, and R. K. Jain, *Numerical Methods for Scientific and Engineering Computation*, ISBN: 81-224-2001-X, 5th Edition, New Age International Publisher, 2007 .
- J. H. Mathews, K. D. Fink, *Numerical Methods Using Matlab*, 4th Edition, ISBN: 9780130652485, Pearson Prentice Hall, 2004.
- B. Hahn, D. Valentine, *Essential Matlab for Engineers and Scientists*, 4th Edition, Elsevier Science, 2009.



برنامه‌سازی پیشرفته (C++) (۴۰۲۴۴) Advanced Programming (C++)

طراح درس: گروه نرم‌افزار*

مقطع: کارشناسی	گرایش: نرم‌افزار
نوع درس: اصلی	واحد: ۳
هم‌نیاز: -	پیش‌نیاز: مبانی برنامه‌سازی

کلیات

هدف از این درس، آشنایی دانشجویان با تکنیک‌های برنامه‌نویسی شیء‌گرا و تکنیک‌های ایجاد برنامه مهندسی‌ساز (UML^۱، TDD^۲ و Refactoring) است.

ریز مواد

خلاصه: یادآوری مفاهیم اولیه و زبان C، کلاس‌ها و شیء‌گرایی، operator overloading، وراثت، چندریختی، template، streaming، پردازش فایل، مدیریت استثناها، کلاس رشته، الگوریتم و ساختمان‌های داده، STL، Refactoring، TDD، گرافیک در C++، چندرسمانی، آشنایی با UML.

۱. یادآوری مفاهیم اولیه و زبان C (۱ جلسه)
 - دستورات کنترلی
 - توابع غیر بازگشتی و بازگشتی
 - اشاره‌گرها
۲. یادآوری انتزاع داده و مفاهیم اولیه کلاس‌ها و شیء‌گرایی (۱ جلسه)
 - structها
 - classها
۳. آرایه‌ها و بردار (vector) (۱ جلسه)
۴. رشته‌ها (۱ جلسه)
 - رشته‌های مبتنی بر اشاره‌گر (رشته در C)
 - کلاس رشته (string): مقداردهی و چسباندن رشته‌ها، مقایسه رشته‌ها، زیررشته‌ها، جابجایی رشته‌ها، جستجو و جایگزینی در رشته‌ها، اضافه کردن کاراکتر در رشته‌ها، تبدیل رشته به *char در رشته
۵. کلاس‌ها و شیء‌گرایی (۳ جلسه)
 - scope کلاس و کلاس‌های دسترسی (private, public, protected)
 - جدا سازی واسط از پیاده‌سازی
 - توابع دسترسی و توابع کاربردی
 - اشیاء constant و اعضای constant و static
 - توابع و کلاس‌های friend
 - اشاره‌گر this

* ریز مواد این درس توسط آقای مهندس ابوالحسن شمسایی تنظیم شده و توسط کمیته بازنگری دروس برنامه‌سازی (منتخب گروه نرم‌افزار) مورد پالایش و تکمیل قرار گرفته است.

¹ Unified Modeling Language

² Test Driven Development

- مدیریت حافظه پویا (new, delete)
- انتزاع داده و پنهان سازی اطلاعات
- کلاس‌های proxy
- ۶. آشنایی با برخی نمودارهای UML (activity diagram, class diagram) (۱ جلسه)
- ۷. مفهوم operator overloading (۲ جلسه)
 - محدودیت‌ها در overload نمودن عملوندها
 - overload کردن stream، تک عملوند، عملوند دوتایی، ++ و --.
 - تبدیلات انواع داده
- ۸. وراثت (۲ جلسه)
 - کلاس پایه و کلاس‌های مشتق
 - اعضای محافظت شده (protected)
 - constructor و destructor در کلاس‌های پایه و مشتق
 - انواع وراثت (private, public, protected)
- ۹. چندریختی (۲ جلسه)
 - چندریختی چیست؟
 - رابطه بین اشیاء در سلسله مراتب وراثت
 - توابع virtual
 - کلاسهای abstract
 - virtual destructor
- ۱۰. برنامه نویسی چند ریسمانی (۲ جلسه)
 - مفهوم ریسمان
 - مفهوم ناحیه بحرانی
 - mutex و conditional variable
 - semaphore
- ۱۱. گرافیک در C++ (۲ جلسه)
 - مفاهیم desktop application programming
 - آشنایی با form، dialog و controlها
 - event handling
- ۱۲. مفهوم template (۱ جلسه)
 - Template در تابعها
 - Overload کردن template در تابعها
 - Template در کلاسها
 - پارامتر nontype و default type
 - Template و inheritance
 - Template و friends
 - Template و static members
- ۱۳. آشنایی با Refactoring و TDD (۲ جلسه)
- ۱۴. مفهوم Streaming (۱ جلسه)
 - آشنایی با کتابخانه و سلسله مراتب وراثت stream
 - آشنایی با stream output
 - آشنایی با stream input
 - آشنایی با I/O بدون فرمت

- آشنایی با دستکاری جریان داده
- تشخیص خطاهای جریان داده
- نوشتن جریان خروجی در جریان ورودی (tie)
- ۱۵. مدیریت استثناها (Exception Handling) (۱ جلسه)
 - مفهوم استثنا
 - ساختار try, catch, finally
 - rethrow کردن یک exception
 - مشخص سازی exception
 - stack unwinding
 - اشیای کلاس exception
- ۱۶. پردازش فایل (۲ جلسه)
 - سلسله مراتب داده
 - رابطه بین فایل و stream
 - خواندن و نوشتن فایل‌های ترتیبی و تصادفی با استفاده از ifstream
 - reinterpret_cast و فایل‌های باینری
- ۱۷. ساختمان‌های داده و الگوریتم جستجو و مرتب‌سازی (۲ جلسه)
 - کلاس‌های خویش-ارجاع
 - تخصیص حافظه بصورت پویا در C++
 - لیست‌های پیوندی، پشته، صف، درخت
 - binary search, insertion sort, selection sort, merge sort
- ۱۸. Standard Template Library (۲ جلسه)
 - Sequence container (vector, deque, list)
 - Associative containers (set, multiset, map, multimap)
 - Container adapters (stack, queue, priority_queue)
- ۱۹. مباحث تکمیلی (۱ جلسه - در صورت وجود فرصت)
 - Introduction to Boost C++ Libraries
 - Difference Between Win C/C++ and Linux C/C++

آزمون - تمرین - پروژه

۱. تمرین مقدمه کلاس‌ها، آرایه، بردار و رشته‌ها
۲. تمرین کلاس، UML و Operator Overloading
۳. تمرین وراثت و چندریختی
۴. تمرین گرافیک، چندریسمانی و Template
۵. تمرین Refactoring و TDD، Streaming و مدیریت استثناها
۶. تمرین فایل، ساختمان داده و الگوریتم، STL و مباحث تکمیلی

نمره از ۲۰	توضیحات	عنوان
۴	حداقل ۴ تمرین	تمرین
۲	حداقل ۴ کوئیز	کوئیز
۴	حداقل ۵ هفته قبل از پایان ترم تعریف شود حداقل مفاهیم شی گرای (کلاس، وراثت و چندریختی)، گرافیک و اسناد UML مربوطه را شامل شود. حداقل از ۲ فاز با مهلت یکی قبل از پایان کلاس درس و دیگری حداکثر تا دو هفته پس از پایان امتحانات تشکیل شود	پروژه
۴	حداقل تا پایان چند ریختی	آزمون میان ترم
۶		آزمون پایانی
(+۱)	مشارکت و فعالیت موثر در کلاس‌های درس و حل تمرین.	فعالیت کلاسی

مرجع اصلی

- P. Deitel, H. Deitel, *C++: How to Program*, 9th Edition, Prentice Hall, 2011.

مراجع فرعی

- J. Soulie, *C++ Language Tutorial*, 2009. Available at <http://www.cplusplus.com/doc/tutorial/>
- M. Fowler, K. Beck, J. Brant, W. Opdyke, D. Roberts, *Refactoring: Improving the Design of Existing Code*, Addison Wesley, 1999.



برنامه‌سازی پیشرفته (جاوا) (۴۰۲۴۴) Advanced Programming (Java)

طراح درس: گروه نرم‌افزار*

مقطع: کارشناسی	گرایش: نرم‌افزار
نوع درس: اصلی	واحد: ۳
هم‌نیاز: -	پیش‌نیاز: مبانی برنامه‌سازی

کلیات

در این درس، مفاهیم برنامه‌نویسی شیء‌گرا در قالب زبان برنامه‌نویسی جاوا تدریس می‌شود. همچنین ویژگی‌های ذاتی زبان جاوا، امکانات برنامه‌سازی در جاوا، تفاوت رویکرد جاوا با زبان‌های مشابه در موارد مختلف، برنامه‌سازی همروند و کیفیت نرم‌افزار مورد توجه قرار می‌گیرد. فرض بر این است که دانشجویان در دروس قبلی، با یک زبان برنامه‌سازی آشنا هستند و برخی روش‌های حل مسأله با کمک برنامه‌سازی را (مانند جستجو، مرتب‌سازی، عملیات ریاضی و ...) می‌شناسند. تأکید در این درس بر مفاهیم شیء‌گراست.

ریز مواد

۱. آشنایی با زبان جاوا (۱ جلسه)
 - تاریخچه
 - ویژگی‌های زبان جاوا
 - اولین برنامه در جاوا
۲. مفاهیم اولیه برنامه‌سازی در جاوا (۳ جلسه)
 - متغیر، متد، شرط، حلقه
 - داده‌های اولیه (Primitive Data Types)
 - رشته
 - آرایه
۳. مقدمه‌ای بر طراحی و برنامه‌سازی شیء‌گرا (۲ جلسه)
 - تاریخچه تکامل رویکردها تا رویکرد شیء‌گرایی
 - مفاهیم اولیه شیء‌گرایی
 - طرز تفکر و طراحی شیء‌گرا
 - لفاف‌بندی، واسط، رده، بسته، دسترسی
۴. برنامه‌سازی شیء‌گرا در جاوا (۶ جلسه)
 - نحوه تعریف کلاس‌ها
 - اشیاء در حافظه و مدیریت حافظه
 - بارگذاری اولیه و مرگ اشیاء در حافظه
 - زباله‌روب (Garbage Collector)
 - روش‌های ارسال پارامتر در زبان‌های مختلف و جاوا
 - سازنده

* ریز مواد این درس توسط آقای مهندس صادق علی اکبری تنظیم شده و توسط کمیته بازنگری دروس برنامه‌سازی (منتخب گروه نرم‌افزار) مورد پالایش و تکمیل قرار گرفته است.

- this
- اعضای ایستا
- بسته (package)
- آشنایی اولیه با نمودار UML Class Diagram
- ۵. وراثت (۲ جلسه)
 - مفهوم وراثت
 - Protected, abstract, super
 - وراثت چندگانه در سایر زبانها
- ۶. چندریختی (۱ جلسه)
 - چندریختی با کمک وراثت
 - مفهوم virtual در زبان C++
 - اعضای final
- ۷. واسط (interface) (۱ جلسه)
 - کاربرد واسط
 - وراثت چندگانه با کمک واسط
- ۸. کیفیت نرم افزار (۲ جلسه)
 - آزمون نرم افزار و آزمون واحد
 - Refactoring
- ۹. ساخت واسط کاربری گرافیکی (۲ جلسه)
 - مقدمه برنامه سازی واسط کاربر
 - اجزای واسط کاربر
 - Event Handler
- ۱۰. مدیریت خطا و استثنا (۲ جلسه)
 - مدل سنتی مدیریت خطا
 - چارچوب مدیریت خطا در جاوا
 - مزایای این مدل
 - Finally, Runtime Exception
- ۱۱. داده های عام (Generics) (۱ جلسه)
 - متدها و کلاس های عام
 - کاربردها
 - تولید و استفاده از کلاس های عام
 - کلاس های عام و وراثت
 - ماهیت کلاس های عام و تفاوت آن ها با Template در زبان C++
- ۱۲. مجموعه ها و ظرفها (Containers) (۲ جلسه)
 - ساختمان داده های موجود در جاوا
 - Collections, ArrayList, LinkedList, Set, Map
 - Iterator
- ۱۳. فایل، جویبار و شبکه (۲ جلسه)
 - ورودی و خروجی در فایل
 - Serialization
 - برنامه سازی تحت شبکه (socket)
- ۱۴. برنامه سازی همروند (۱ جلسه)
 - نیاز به همروندی

- همروندی در جاوا
- طول عمر یک thread
- آشنایی اولیه با مفاهیم Synchronization و critical section
- ۱۵. مفاهیم پیشرفته و تکمیلی (۱ جلسه)
 - کلاس‌های داخلی و کلاس‌های بی‌نام
 - حاشیه‌گذاری (Annotation)
 - Enumeration
 - ۱۶. Reflection (۱ جلسه)
 - نیاز به RTTI
 - RTTI در جاوا
 - کاربردها

آزمون - تمرین - پروژه

نمره از ۲۰	تعداد	موضوع
۴	۱۶	کوئیز
۳	۱	آزمون میان‌ترم
۵	۱	آزمون پایان‌ترم
۳	۳	تمرین
۵	۳	پروژه

تمرین‌ها به صورت مستقل تعریف می‌شوند و هر یک بخش‌هایی از دانش برنامه‌سازی دانشجو را محک می‌زنند. اما پروژه، یک برنامه مشخص است که در سه فاز در طول ترم تکمیل می‌شود. موضوع پروژه باید به گونه‌ای انتخاب شود که در آن‌ها اشیاء مختلف ظهور پیدا کنند و نیاز به طراحی شیء‌گرا احساس شود. در فاز اول، طراحی کلی اشیاء و ساختار پروژه انجام می‌شود. در فاز دوم برنامه‌سازی واسط کاربر اضافه می‌شود و ساختار طراحی کامل می‌شود. در فاز سوم، که حجم بیشتری دارد، بخش‌های باقیمانده از نیازمندی‌های پروژه پیاده‌سازی می‌شود. در فاز سوم از مفاهیم پیشرفته درس مانند استفاده از فایل و برنامه‌سازی همروند نیز استفاده می‌شود. زمانبندی و مواد تمرین‌ها و پروژه‌ها در جدول زیر آمده است:

عنوان	مهلت تحویل	موضوع
تمرین ۱	جلسه ۸	شروع برنامه‌سازی با جاوا، کار با انواع داده اولیه، کار با رشته‌ها، کار با آرایه‌ها، توانایی حل مسأله
تمرین ۲	جلسه ۱۴	مفاهیم اولیه شیء‌گرایی
پروژه - فاز ۱	جلسه ۲۰	تعریف مسأله، طراحی و پیاده‌سازی ساختار اولیه پروژه
تمرین ۳	جلسه ۲۴	تست، Refactoring، مفاهیم پیشرفته در شیء‌گرایی
پروژه - فاز ۲	جلسه ۳۰	برنامه‌سازی واسط کاربر، تکمیل طراحی و پیاده‌سازی
پروژه - فاز ۳	یک هفته بعد از پایان امتحانات	تکمیل نیازمندی‌های پروژه، استفاده از مفاهیم پیشرفته زبان

مرجع اصلی

- P. Deitel and H. Deitel, *Java: How to Program*, 9th Edition, Prentice Hall, 2011.

- B. Eckel, *Thinking in Java*, 4th Edition, Prentice Hall, 2006.
- M. Fowler, K. Beck, J. Brant, W. Opdyke, D. Roberts, *Refactoring: Improving the Design of Existing Code*, Addison-Wesley, 1999.



ساختمان داده‌ها و الگوریتم‌ها (۴۰۲۵۴)

Data Structures and Algorithms

طراح درس: محمد قدسی

مقطع: کارشناسی	گرایش: نرم‌افزار
نوع درس: اصلی	واحد: ۳
هم‌نیاز: برنامه‌سازی پیشرفته	پیش‌نیاز: ساختمان‌های گسسته

کلیات

موضوع این درس در واقع مبانی داده‌ساختارها و الگوریتم‌هاست که در آن شما با داده‌ساختارهای ساده ولی مهم و نیز با الگوریتم‌های مقدماتی آشنا می‌شوید. در ارائه‌ی مطالب این درس بر تحلیل و اثبات درستی الگوریتم‌ها تاکید می‌شود.

مطالبی که دانشجوی باید از قبل بداند:

- آشنائی کامل با یکی از زبان‌های برنامه‌نویسی C++ یا Java
- روش‌های بازگشتی در حل مسایل، استفاده از استقراء.

البته الگوریتم‌های درس مستقل از زبان و مطابق دستورهای کتاب CLRS گفته می‌شود اما توصیه می‌شود که تمرین‌های برنامه‌نویسی را به زبان جاوا بنویسید.

ریز مواد

۱. معرفی درس
 - a. سطوح انتزاع
 - b. مراحل مختلف حل مسئله و انتزاع
 - c. داده‌مدل‌ها، داده‌گونه‌ها، داده‌ساختارها، داده‌گونه‌ی انتزاعی، شیء
۲. روش‌های تحلیل الگوریتم‌ها: تابع رشد، شمارش مراحل، رابطه‌های بازگشتی و روش‌های حل آن‌ها (حدس و استقراء، تکرار با جای‌گذاری و استفاده از قضیه‌ی اصلی)، رابطه‌های همگن و تحلیل سرشکنی
۳. داده‌ساختارهای لیست
 - a. تعاریف اولیه، عملیات بر روی لیست‌ها
 - b. داده‌ساختارها برای لیست‌ها، کار با اشاره‌گرهای واقعی و اندیسی
 - c. انواع لیست‌ها: صف، پشته، پیاده‌سازی آن‌ها، لیست‌های کلی
 - d. مسائل: مرتب‌سازی ادغامی و تبدیل الگوریتم‌های بازگشتی به غیربازگشتی و زباله رویی
۴. داده‌ساختارهای درخت
 - a. تعاریف اولیه، درخت عبارت
 - b. پیاده‌سازی مختلف درخت‌ها، استقراء بر روی درخت، پیمایش درخت‌ها
 - c. استقرای ساختاری، درخت دودویی
 - d. اعمال مختلف بر روی درخت عبارت، تبدیل نگارش‌های مختلف یک عبارت ریاضی
 - e. ترای برای ذخیره‌ی رشته‌ها
 - f. درخت دودویی جست‌وجو
 - g. صف اولویت (هرم بیشینه و کمینه)
۵. روش درهم‌سازی

۶. مرتب‌سازی و مرتبه‌ی آماری

- a. کران پایین این الگوریتم‌ها
- b. درخت تصمیم
- c. مرتب‌سازها با مرتبه‌ی خطی (شمارشی، مبنایی و سطلی)
- d. مرتب‌سازی سریع
- e. مرتب‌سازی هرمی
- f. مرتبه‌ی آماری
- g. مرتب‌سازی خارجی

۷. داده‌ساختارهای پیشرفته

- a. مجموعه‌های مجزا
- b. درخت دودویی جست‌وجوی بهینه
- c. درخت‌های دودویی جست‌وجوی متوازن
 - i. درخت قرمز-سیاه»
 - ii. درخت مرتبه‌ی آماری
 - iii. درخت بازه
 - iv. درخت ا.وی.ال
 - v. درخت‌های ۲-۳ و بی

آزمون - تمرین

در این درس تقریباً هر دو هفته یک تمرین کتبی و هر سه هفته یک تمرین برنامه‌نویسی داده می‌شود. شما باید تمرین‌های کتبی را حل کنید ولی تحویل آن‌ها ضروری نیست. در آزمون‌های کوتاه، تعدادی از مسئله‌های این تمرین‌ها انتخاب می‌شوند که حل کنید.

- ۲ تمرین کتبی که باید تحویل دهید (۱ نمره)
- ۳ آزمون کوتاه از مسائل داده شده در تمرین‌ها (۳ نمره)
- ۴ تمرین برنامه‌نویسی (۳ نمره)
- ۲ عدد آزمون میان‌ترم (۷ نمره)
 - آزمون بهتر ۴ نمره خواهد داشت.
- آزمون نهایی (۶ نمره)

توجه: ملاک قبولی در این درس نمره‌های آزمون‌هاست. شرط قبولی کسب حداقل ۸ نمره از ۱۶ نمره این آزمون‌هاست.

اخطار نسبت به تقلب: آن‌چه شما تحویل می‌دهید (برنامه، حل تمرین و آزمون‌ها) باید دقیقاً کار خودتان باشد. البته گرفتن مشورت قبل از تهیه‌ی مدارک ایرادی ندارد، اما کار تحویلی باید فقط کار شما باشد. در صورت کشف کپی، نمره‌ی دو طرف صفر خواهد شد، و در صورت تکرار (حداقل یک بار دیگر)، متخلف از درس نمره‌ی قبولی نخواهد گرفت.

کلاس‌های تمرین: شرکت در کلاس‌های تمرین این درس با توجه به نیاز به فراگیری استفاده از جاوا در طراحی شی‌گرای داده‌ساختارها ضروری است.

حل تمرین: اکیداً توصیه می‌شود که با نرم‌افزار FarsiTeX تمرین‌های خود را حل کنید و لازم است آن را به آدرسی که داده می‌شود ایمیل کنید، در این مورد کلاسی ارائه خواهد شد.

مراجع اصلی

- محمد قدسی، داده ساختارها و الگوریتم‌ها، انتشارات فاطمی، شهریور ۱۳۸۸. (مرجع اصلی)
- T. Cormen, C. Leiserson, R. Rivest, C. Stein (CLRS), *Introduction to Algorithms*, MIT Press, Sept. 2001.



مقطع: کارشناسی	گرایش: نرم‌افزار
نوع درس: اصلی	واحد: ۳
هم‌نیاز: -	پیش‌نیاز: ساختمان داده‌ها و الگوریتم‌ها

کلیات

در این درس روش‌های مختلف طراحی و تحلیل الگوریتم‌های سریع و کارا برای حل مسایل مختلف گفته می‌شوند. در ارائه‌ی مطالب بر استفاده از داده‌ساختارهای مناسب و اثبات درستی و تحلیل الگوریتم‌ها تاکید می‌شود.

ریز مواد

۱. مقدمات
۲. (تکرار) تحلیل سرشکنی amortized analysis
۳. روش‌های طراحی الگوریتم‌ها
 - (a) بر اساس استقراء
 - (b) تقسیم و حل (فرش کردن، تورنمنت بازی‌ها، ضرب چندجمله‌ای‌ها، آسمان خراش‌ها، شبکه‌های مرتب‌ساز، الگوریتم استراسن)
 - (c) پویا (ضرب ماتریس‌ها، مسایل کوله‌پشتی، مثلث‌بندی، LCS، پاراگراف‌بندی)
 - (d) حریصانه (مسایل زمان‌بندی، خردکردن پول، کد هافمن)
 - (e) جست‌وجوی کامل: روش پس‌گرد (backtracking)، درخت بازی، هرس α - β ، روش انشعاب‌وحد (Branch-and-Bound)، مسئله‌ی فروشنده‌ی دوره‌گرد.
۴. الگوریتم‌های گراف
 - (a) جست‌وجوی گراف به‌صورت عمق-اول، سطح-اول
 - (b) مرتب‌سازی توپولوژیکی (topological)، پیداکردن دور، اجزای همبند، اجزای دوهمبند
 - (c) درخت پوشای کمینه (الگوریتم‌های Prim و Kruskal)
 - (d) کوتاه‌ترین مسیرها در گراف: الگوریتم‌های Bellman-Ford، Dijkstra و Floyd
 - (e) شبکه‌های شار (الگوریتم‌های Edmond-Karp و Ford-Fulkerson)
۵. تطابق رشته‌ها (string matching)
۶. مسایل ان‌پی-تمام: مقدمات، نظریه‌ی ان‌پی-تمام، مسایل اصلی (3-Sat، 3D-matching، Vertex-Cover، Clique، دور همیلتونی، افراز) روش‌های اثبات ان‌پی-تمام بودن یک مسئله، استفاده از این موضوع برای تحلیل الگوریتم‌ها.

آزمون - تمرین

در این درس تقریباً هر دو هفته یک تمرین کتبی و هر سه هفته یک تمرین برنامه‌نویسی داده می‌شود. شما باید تمرین‌های کتبی را حل کنید ولی تحویل همه‌ی آن‌ها ضروری نیست. در آزمونک‌ها، تعدادی از مسئله‌های این تمرین‌ها انتخاب می‌شوند که حل کنید.

- ۲ تمرین کتبی که باید تحویل دهید (۱ نمره)
- ۳ آزمونک از مسائل داده شده در تمرین‌ها (۳ نمره)
- ۴ تمرین برنامه‌نویسی (۴ نمره)

- یک مسابقه برنامه‌نویسی به سبک ای‌سی‌ام (۱ تا ۲ نمره)
 - بسته به رتبه تا حداکثر ۱ نمره اضافه داده خواهد شد.
 - ۲ عدد آزمون میان‌ترم (۷ نمره)
 - آزمون بهتر ۴ نمره خواهد داشت.
 - آزمون نهایی (۶ نمره)
- توجه: ملاک قبولی در این درس نمره‌های آزمون‌هاست. شرط قبولی کسب حداقل ۸ نمره از ۱۶ نمره این آزمون‌هاست.

مرجع اصلی

- T. Cormen, C. Leiserson, R. Rivest, C. Stein (CLRS) *Introduction to Algorithms*, MIT Press, Sept. 3rd Edition, 2009.

مراجع کمکی

- J. Kleinberg, E. Tardos, *Algorithm Design*, Addison Wesley, 2005.
- محمد قدسی و محمد مهدیان، مسئله‌های الگوریتمی، انتشارات فاطمی، ۱۳۸۷.
- U. Manber, *Introduction to Algorithms: A Creative Approach*, Addison Wesley, 1989.

مبانی مدارهای الکتریکی و الکترونیکی (۴۰۱۲۴) Fundamentals of Electrical and Electronic Circuits



دانشکده مهندسی کامپیوتر

طراح درس: علی محمدافشین همت یار

مقطع: کارشناسی	گرایش: نرم افزار
نوع درس: اصلی	واحد: ۳
هم‌نیاز: معادلات دیفرانسیل	پیش‌نیاز: فیزیک ۲

کلیات

این درس، حاوی مطالب اساسی و مهم در زمینه مدارهای الکتریکی و الکترونیکی است که به طور خاص برای دانشجویان نرم‌افزار طراحی شده است.

ریز مواد

- (۱) قوانین و تعاریف (۶ ساعت)
 - قوانین ولتاژ و جریان
 - عناصر یک‌دریجه ایده‌آل و واقعی (مقاومت، خازن، سلف و منابع ولتاژ و جریان وابسته)
 - عناصر دودریجه (منابع ولتاژ و جریان وابسته، و تقویت‌کننده عملیاتی)
 - مفاهیم توان و انرژی و عناصر فعال و غیرفعال
 - شکل موج‌ها (پله، پالس، ضربه و سینوسی)
- (۲) تحلیل مدارها در حوزه زمان (۹ ساعت)
 - روش‌های تجزیه و تحلیل گره و مش
 - معادله دیفرانسیل برای مدارهای ساده، مرتبه یک و مرتبه دو
 - پاسخ‌های حالت صفر، ورودی صفر، حالت گذرا و حالت دائمی
 - پاسخ پله و پاسخ ضربه
- (۳) تحلیل مدارها در حوزه فرکانس (۶ ساعت)
 - پاسخ حالت دائمی سینوسی و تابع شبکه
 - تحلیل مدار با استفاده از تبدیل لاپلاس
- (۴) قضایای شبکه و کاربرد آنها (۶ ساعت)
 - قضیه جمع آثار
 - قضایای تونن و نورتن
- (۵) دیود معمولی، دیود زبر و کاربردهای آنها (۶ ساعت)
 - مشخصه و مدل دیود معمولی و دیود زبر
 - یکسوسازها و محدودکننده‌های ولتاژ
- (۶) ترانزیستورهای MOSFET (۳ ساعت)
 - مشخصات و محدودیت‌های عملی مدارهای منطقی
 - منحنی‌های مشخصه و مدل ترانزیستورهای MOSFET
- (۷) خانواده‌های منطقی با ترانزیستورهای MOSFET (۹ ساعت)

- مبانی و انواع مدارهای منطقی خانواده NMOS
- مبانی و انواع مدارهای منطقی خانواده CMOS
- (۸) مدارهای کاربردی (۶ ساعت)
- مولتی‌ویبراتورها
- مبدل‌های آنالوگ به دیجیتال و دیجیتال به آنالوگ

آزمون - تمرین

- دو آزمون میان ترم (۳۰٪)
- آزمون پایان ترم (۴۰٪)
- کوئیزها (۱۰٪)
- تمرینات (۲۰٪)

مراجع اصلی

- W. H. Hayt, J. E. Kemmerly, S. M. Durbin, *Engineering Circuit Analysis*, 6th Edition, McGraw Hill, 2002.
- R. Boylestad, L. Nashelsky, *Electronic Devices and Circuit Theory*, 4th Edition, Prentice Hall, 1987.
- T. A. DeMassa, Z. Ciccone, *Digital Integrated Circuits*, Wiley, 1996.
- عادل صدره و کنت اسمیت، مدارهای میکروالکترونیک (ویراست چهارم)، نشر علوم دانشگاهی، ۱۳۸۱.



بازیابی پیشرفته اطلاعات (۴۰۳۲۴) Advanced Information Retrieval

طراح درس: حسن ابوالحسنی

مقطع: کارشناسی	گرایش: نرم افزار
نوع درس: اصلی	واحد: ۳
هم‌نیاز: -	پیش‌نیاز: طراحی الگوریتم‌ها

کلیات

هدف این درس، آشنا نمودن دانشجویان کارشناسی با مفاهیم جدید در حوزه بازیابی اطلاعات با تاکید بر موتورهای جستجوی وب است. یک بخش عمده این درس معطوف به سه مدل بازیابی پایه یعنی مدل بولی، مدل برداری و مدل احتمالی در بازیابی اطلاعات است. در کنار آشنایی با این مدل‌ها داده ساختارهای اساسی برای پیاده‌سازی یک موتور جستجو نیز مورد بحث قرار می‌گیرد. ارزیابی سیستم‌های بازیابی اطلاعات، تغییر و گسترش پرسمان، بازیابی ایکس ام ال، دسته‌بندی اسناد و خوشه‌بندی اسناد از مطالب دیگر درس است. یک بخش مهم دیگر این درس مطالعه خصوصیات وب، خزنده‌های وب و مفهوم تحلیل لینک است.

ریز مواد

۱. مقدمه - معرفی بازیابی اطلاعات و مدل بولی (۱ جلسه)
۲. مراحل ساخت دیکشنری لغات (۱ تا ۲ جلسه)
۳. بازیابی منعطف (۱ تا ۲ جلسه)
۴. ساخت اندیس (۱ تا ۲ جلسه)
۵. فشرده‌سازی اندیس (۱ تا ۲ جلسه)
۶. مدل برداری (۲ تا ۳ جلسه)
۷. نحوه ارزیابی سیستم‌های بازیابی اطلاعات (۱ تا ۲ جلسه)
۸. توسعه پرسمان (۱ تا ۲ جلسه)
۹. بازیابی اسناد ایکس ام ال (۱ تا ۲ جلسه)
۱۰. مدل احتمالی (۲ تا ۳ جلسه)
۱۱. دسته‌بندی اسناد (۲ تا ۳ جلسه)
۱۲. خوشه‌بندی اسناد (۲ تا ۳ جلسه)
۱۳. مشخصه‌های وب (۱ تا ۲ جلسه)
۱۴. معماری خزنده‌های وب (۲ تا ۳ جلسه)
۱۵. تحلیل پیوند در وب (۲ تا ۳ جلسه)

آزمون - تمرین

- آزمون: آزمون‌های میان ترم و پایان ترم (۷۰٪ از کل نمره)
- تمرین: سه تا چهار تمرین عملی در طول نیم‌سال تحویل داده می‌شوند (۲۰٪ از کل نمره)
- تکلیف: چهار تا پنج تکلیف در طول نیم‌سال تحویل داده می‌شوند (۱۰٪ از کل نمره)

- C. D. Manning, P. Raghavan, H. Schutze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.
- R. Baeza-Yates, E. Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999.



هوش مصنوعی (۴۰۴۱۷)

Artificial Intelligence

طراح درس: غلامرضا قاسم ثانی

مقطع: کارشناسی	گرایش: نرم افزار
نوع درس: اصلی	واحد: ۳
هم نیاز: -	پیش نیاز: طراحی الگوریتمها

کلیات

هوش مصنوعی یکی از شاخه‌های مهم علوم کامپیوتر است که هدف نهایی آن مدل‌سازی هوش و عملکرد هوشمندانه انسانها است. هوش مصنوعی خود شامل طیف گسترده‌ای از زیرشاخه‌ها شامل موضوعات کلی نظیر یادگیری ماشینی و انواع درک مفاهیم تا موضوعات خاص‌تر نظیر بازی شطرنج، اثبات خودکار قضایای ریاضی و یا تشخیص بیماری‌ها می‌شود. در این درس ضمن ارائه مفاهیم پایه هوش مصنوعی شامل روشهای مختلف جستجو و نمایش دانش، برخی از زیرشاخه‌های هوش مصنوعی نیز بطور اجمالی معرفی می‌شود. همچنین یکی از زبان‌های برنامه‌سازی ویژه پیاده‌سازی کاربردهای هوش مصنوعی نظیر زبانهای پرولوگ و یا لیسپ معرفی می‌شود.

ریز مواد

۱. مقدمه و تاریخچه (۲ جلسه)
۲. مامورین هوشمند (۲ جلسه)
 - a. انواع محیط مساله
 - b. انواع ساختار مامورین هوشمند
۳. حل مساله به کمک جستجو
 - a. جستجوی سیستماتیک (۲ جلسه)
 - i. عمق اول و سطح اول
 - ii. عمق اول با تعمیق مکرر
 - iii. انشعاب و حد
 - b. جستجوی ابتکاری (۶ جلسه)
 - i. تولید و آزمون
 - ii. تپهنوردی و ذوب فلز شبیه‌سازی شده
 - iii. بهترین اول: A*, AO*, IDA*, RTA*, Beam Search
۴. مسائل ارضاء محدودیت و روشهای حل آنها (۲ جلسه)
 - a. عقب‌گرد ساده (BT)
 - b. روش‌های آینده‌نگر (FC, AC و PC)
 - c. عقب‌گرد هوشمند (BJ)
۵. روشهای جستجوی ویژه بازی (۱ جلسه)
 - a. روش کمینه، بیشینه
 - b. هرس آلفا-بتا
 - c. روشهای مبتنی بر شانس
۶. روش‌های نمایش دانش (۶ جلسه)
 - a. منطق گزاره‌ها
 - b. منطق مسندات مرتبه‌ی اول

- c. فرم‌های نرمال منطق
- d. یکسان‌سازی و رزولوشن
- e. شبکه‌های معنایی وقاب‌ها
- f. استدلال غیریکنواخت
- g. سیستم نگهدارنده درستی
- ۷. معرفی زبان پرولوگ یا لیسپ (۳ جلسه)
- ۸. معرفی مقدماتی زیرشاخه‌های هوش مصنوعی (۵ جلسه)
- a. برنامه‌ریزی در هوش مصنوعی
- b. پردازش زبان‌های طبیعی

آزمون - تمرین - پروژه

- آزمون میان ترم (/۳۵)
- آزمون پایان ترم (/۳۵)
- پروژه عملی (/۲۰)
- کوئیزها و تمرینات (/۱۰)

مرجع اصلی

- S. Russel, P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd Edition, Prentice Hall, 2003.

مراجع کمکی

- I. Bratko, *Prolog Programming for Artificial Intelligence*, 3rd Edition, Addison Wesley, 2001.
- N. J. Nilsson, *Artificial Intelligence: A new Synthesis*, Morgan Kaufmann, 1998.
- E. Rich, and K. Knight, *An Introduction to Artificial Intelligence*, McGraw-Hill, 1991.



نظریه زبان‌ها و ماشین‌ها (۴۰۴۱۵)

Theory of Machines and Languages

طراح درس: علی موقر رحیم آبادی

مقطع: کارشناسی	گرایش: نرم‌افزار
نوع درس: اصلی	واحد: ۳
هم‌نیاز: -	پیش‌نیاز: ساختمان داده‌ها و الگوریتم‌ها

کلیات

این درس درباره جنبه‌های نظری رشته مهندسی و علوم کامپیوتر است. مباحث مورد بررسی شامل مدل‌های مختلف محاسباتی، توانایی محاسباتی این مدل‌ها، خواص محاسباتی آنها و کاربردهای آنها می‌باشد. دیگر مباحث شامل مفاهیم محاسبه پذیری، تصمیم پذیری و تز چرچ و تورینگ در مورد الگوریتم‌هاست.

ریز مواد

۱. مباحث مقدماتی (۴ جلسه)
منطق گزاره‌ای، منطق مسندی، سیستم اثبات، نظریه مجموعه‌ها، پارادوکس راسل، مجموعه‌های شمارا و ناشمارا، زبان‌ها و گرامر‌ها.
۲. ماشین‌های حالت متناهی (۸ جلسه)
پذیرنده‌های متناهی قطعی، پذیرنده‌های متناهی غیرقطعی، زبان‌های منظم، عبارات منظم، گرامرهای راستگرد خطی، گرامرهای چپگرد خطی، گرامرهای منظم، گرامرهای خطی، زبان‌های نامنظم، لیم پمپینگ برای زبان‌های منظم.
۳. زبان‌های مستقل از متن (۱۲ جلسه)
گرامرهای مستقل از متن، زبان‌های مستقل از متن، اشتقاق چپگرد، اشتقاق راستگرد، درخت اشتقاق، گرامرهای مبهم، گرامرهای نامبهم، زبان‌های ذاتاً مبهم، زبان‌های نامبهم، ساده‌سازی گرامرهای مستقل از متن، گرامرهای مستقل از متن به صورت طبیعی چامسکی، گرامرهای مستقل از متن به صورت طبیعی گرایباخ، مسأله عضویت، الگوریتم CYK، ماشین‌های پوش دان، هم‌ارزی ماشین‌های پوش دان و گرامرهای مستقل از متن، ماشین‌های پوش دان قطعی، زبان‌های مستقل از متن قطعی، زبان‌های غیر مستقل از متن، لیم پمپینگ برای زبان‌های مستقل از متن.
۴. محاسبه پذیری (۶ جلسه)
ماشین تورینگ، تز چرچ و تورینگ، تصمیم پذیری و تصمیم ناپذیری، محاسبه پذیری و محاسبه ناپذیری، مسئله توقف، مسئله تخصیص پست، پیچیدگی محاسباتی، رده پیچیدگی P، رده پیچیدگی NP، مسائل NP کامل، مسائل NP سخت.

آزمون - تمرین

- تمرینات هفتگی (۳۰٪ نمره کل)
- کوئیزها (۴۵٪ نمره کل)
- آزمون پایان نیمسال (۲۵٪ نمره کل)

مرجع اصلی

- M. Sipser, *Introduction to the theory of computation*, 2nd Edition, PWS Publishing Company, 2006.

- P. Linz, *An introduction to formal languages and automata*, 3rd Edition, Jones and Barlett Publishers, 2001.
- J.E. Hopcroft, R. Motwani, J.D. Ullman, *Introduction to automata theory, languages, and computaion*, 2nd Edition, Addison Wesley, 2001.
- J. P. Denning, J. B. Dennis, J. E. Qualitz, *Machines, languages, and computaion*, Prentice Hall, 1978.
- J. E. Hopcroft, J.D. Ullman, *Introduction to automata theory, languages, and computaion*, Addison Wesley, 1979.
- P. J. Cameron, *Sets, Logics and Categories*, Springer Verlag, 1998.



سیستم‌های عامل (۴۰۴۲۴) Operating Systems

طراح درس: حمید بیگی

مقطع: کارشناسی	گرایش: نرم‌افزار
نوع درس: اصلی	واحد: ۳
هم‌نیاز: -	پیش‌نیاز: معماری کامپیوتر

کلیات

هدف از این درس، آشنا نمودن دانشجویان کارشناسی با اصول سیستم‌های عامل است. در این درس ضمن معرفی بخش‌های مختلف یک سیستم‌عامل و معماری‌های متداول آن، معماری چندین سیستم‌عامل نیز بیان می‌گردد. در این درس تلاش می‌شود تا نمونه‌های مختلفی از سیستم‌های عامل از شرکت‌های مختلف مطرح شود. در این میان سیستم‌های عامل Z/OS شرکت IBM، سیستم‌عامل Solaris شرکت SUN، سیستم‌عامل MacOS شرکت Apple، Linux، و بعضی نسخه‌های Unix و سیستم‌عامل Symbian مطرح شود. این درس دارای چند پروژه برنامه نویسی است که در آنها مفاهیم اولیه سیستم‌های عامل آموزش داده می‌شود.

ریز مواد

۱. بخش نخست: مقدمه ای بر معماری سیستم‌های عامل (۳ جلسه)
 - سیستم‌های عامل، ساختار و اجزای تشکیل دهنده آن (۲ جلسه)
 - معرفی ماشین‌های مجازی از قبیل JAVA, VMware, IBM ZVM (۱ جلسه)
۲. بخش دوم: مدیریت فرایندها (۱۲ جلسه)
 - فرایندها، برنامه ریزی و شیوه ارتباط بین آنها (۲ جلسه)
 - نخ‌ها، مدل‌های پیاده‌سازی نخ‌ها و معرفی کتابخانه‌های مربوط به نخ جاوا و Pthread (۲ جلسه)
 - برنامه ریزی پردازنده (۱ جلسه)
 - پیاده‌سازی بخش مدیریت فرایند و مدیریت منابع یک سیستم‌عامل کوچک (۲ جلسه)
 - همگام‌سازی فرایندها، معرفی نواحی بحرانی، معرفی روش‌های سخت‌افزاری و نرم‌افزاری همگام‌سازی شامل سمافور، مانیتور و عبارت‌های مسیر (۴ جلسه)
 - بن بست (۱ جلسه)
۳. بخش سوم: مدیریت حافظه (۸ جلسه)
 - مدیریت حافظه، شامل: افزاز حافظه، صفحه‌بندی، قطعه‌بندی، ترکیب صفحه‌بندی و قطعه‌بندی، معرفی بخش مدیریت حافظه چند پردازنده از قبیل: پنتیوم، UltraSparc، IBM Z System (۴ جلسه)
 - مدیریت حافظه مجازی، شامل: درخواست صفحه، جایگزینی صفحه، تخصیص حافظه سیستم‌عامل، Thrashing و معرفی بخش مدیریت حافظه چند سیستم‌عامل (۴ جلسه)
۴. بخش چهارم: مبانی محافظت و امنیت (۱ جلسه)
 - مبانی محافظت و امنیت سیستم‌های عامل (۱ جلسه)
۵. بخش پنجم: مبانی سیستم‌عامل‌های بی‌درنگ (۱ جلسه)
 - مبانی سیستم‌عامل‌های بی‌درنگ، مدیریت پردازنده پردازنده و معرفی چند سیستم‌عامل بی‌درنگ نمونه (۱ جلسه)
۶. بخش ششم: مدیریت حافظه جانبی (۴ جلسه)
 - واسطه سیستم‌فایل (۱ جلسه)

- پیاده سازی سیستم فایل (۱ جلسه)
 - حافظه های جانبی و برنامه ریزی دیسک (۱ جلسه)
 - زیر سیستم ورودی و خروجی سیستم عامل (۱ جلسه)
۷. بخش هفتم : مباحث تکمیلی (۱ جلسه)

آزمون - تمرین - پروژه

- آزمون های میان ترم و پایان ترم (۶۰٪ از کل نمره)
- ۵ تمرین که در طول نیم سال تحویل داده می شوند (۱۵٪ از کل نمره)
- ۴ پروژه برنامه نویسی که در طول نیم سال تحویل داده می شوند (۲۵٪ از کل نمره)

مراجع اصلی

- P. Silberschatz, B. Galvin, G. Gagne, *Operating System Concepts*, 8th Edition, John Wiley, 2010.
- R. Elmasri, A. G. Carrick, D. Levine, *Operating Systems: A Spiral Approach*, Mcgraw-Hill, 2009.

توجه: برای مثال هایی که از پردازنده و سیستم های عامل مختلف در درس بیان می شود از کتاب های مختلفی استفاده می گردد که به صورت موردی در کلاس معرفی خواهند شد.



آزمایشگاه سیستم‌های عامل (۴۰۴۰۸) Operating Systems Laboratory

طراح درس: حمید بیگی

مقطع: کارشناسی	گرایش: نرم‌افزار
نوع درس: اصلی	واحد: ۱
هم‌نیاز: -	پیش‌نیاز: سیستم‌های عامل

کلیات

هدف از این آزمایشگاه، آموزش بخش‌های مختلف سیستم عامل لینوکس، استفاده از این بخش‌ها و پیاده‌سازی الگوریتم‌هایی در هر کدام از این بخش‌ها می‌باشد. پس از گذراندن این آزمایشگاه، دانشجویان با ساختار سیستم عامل لینوکس آشنا خواهند شد و توانایی تغییر و کامپایل آن را خواهند داشت. کلیات این آزمایشگاه بصورت زیر می‌باشد اما جزئیات هر آزمایش از یک نیم‌سال به نیم‌سال دیگر تغییر خواهد یافت. در طول یک نیم‌سال ممکن است همه بخش‌های سرفصل پوشش داده نشود اما در همه نیم‌سال‌ها آزمایش‌های ابتدایی پوشش داده خواهد شد و سپس بر روی موضوع‌های مختلفی تمرکز خواهد شد.

ریز مواد

۱. کامپایل و نصب لینوکس
۲. برنامه‌نویسی با زبان ++C و shell در لینوکس
۳. بکارگیری System Call های لینوکس در برنامه‌ها
۴. بررسی رفتار سیستم عامل (مسیر /proc)
۵. ایجاد، اجرا و ازبین بردن فرایندها و نخ‌ها (بکارگیری کتابخانه pthread)
۶. همگام‌سازی و ارتباط بین فرایندها و نخ‌ها (Process & Thread synchronization and inter-process communication)
۷. مدیریت حافظه، حافظه مشترک و حافظه مجازی (Memory, Shared Memory, and Virtual Memory Management)
۸. برنامه‌ریزی پردازنده (CPU Scheduling)
۹. بکارگیری سیستم‌های فایل قابل نصب (Installable File Systems)
۱۰. برنامه‌ریزی دیسک و دستگاه‌های ورودی و خروجی (IO Scheduling)
۱۱. طراحی و پیاده‌سازی Device Driver
۱۲. بکارگیری از سازوکارهای امنیتی لینوکس
۱۳. آشنایی با سیستم‌عامل‌های بی‌درنگ و نهفته
۱۴. آشنایی با Windows Research Kernel

مراجع اصلی

- P. J. Salzman, M. Burian, O. Pomerantz, *The Linux Kernel Module Programming Guide*, 2007.
- K. Wall, M. Watson, M. Whitis, *Linux Programming Unleashed*, Macmillan Computer Publishing, 1999.
- M. Mitchell, J. Oldham, A. Samuel, *Advanced Linux Programming*, New Rivers, 2001.
- C. S. Rodriguez, G. Fischer, S. Smolski, *The Linux® Kernel Primer: A Top-Down Approach for x86 and PowerPC Architectures*, Prentice Hall, 2005.
- J. Corbet, A. Rubini, G. Kroah-Hartman, *Linux Device Drivers*, O'Reilly Books, 2005.



طراحی کامپایلرها (۴۰۴۱۴) Compiler Design

طراح درس: غلامرضا قاسم ثانی

مقطع: کارشناسی	گرایش: نرم افزار
نوع درس: اصلی	واحد: ۳
هم نیاز: -	پیش نیاز: نظریه زبانها و ماشینها

کلیات

طراحی و ساخت کامپایلرها یکی از مفاهیم بنیادی علوم کامپیوتر است. علیرغم آنکه روشهای ساخت کامپایلرها تنوع کمی دارند، لیکن می توانند برای ساخت مفسرها و مترجمهای طیف گسترده و متنوعی از زبانها و ماشینها استفاده شوند. در این درس موضوع ساخت کامپایلرها از طریق توصیف مولفه های اصلی یک کامپایلر، وظایف و ارتباط آنها معرفی می شود. پس از معرفی مقدماتی درباره ی اجزاء یک کامپایلر و انواع گرامرها، مراحل مختلف ترجمه از قبیل تجزیه و تحلیل لغوی، نحوی و معنایی و تولید و پرداخت کد تشریح می شود.

ریز مواد

۱. مقدمه (۲ جلسه)
۲. انواع زبانها و گرامرها (۱ جلسه)
۳. تحلیل واژه های و اصلاح خطاهای واژه ای (۳ جلسه)
۴. تحلیل نحوی
 - روشهای تجزیه بالا به پایین (۵ جلسه)
 - تجزیه پایینگرد
 - تجزیه LL(1)
 - برخورد با خطاهای نحوی
 - روشهای تجزیه پایین به بالا (۸ جلسه)
 - تقدم عملگر
 - تقدم ساده
 - تجزیه LR(1) شامل SLR(1), LALR(1) و CLR(1)
۵. تحلیل معنایی (۱ جلسه)
۶. مدیریت جدول علائم (۱ جلسه)
۷. روشهای تخصیص حافظه ی زمان اجرا (۲ جلسه)
۸. تولید کد (۵ جلسه)
۹. پرداخت و بهینه سازی کد (۱ جلسه)
۱۰. تولید خودکار کامپایلرها (۱ جلسه)

آزمون - تمرین - پروژه

- آزمون میان ترم (۳۵٪)
- آزمون پایان ترم (۳۵٪)
- پروژه عملی (۲۰٪)
- کوئیزها و تمرینات (۱۰٪)

مرجع اصلی

- A. Aho, M. Lam, R. Sethi, J. Ullman, *Compilers: Principles, Techniques, and Tools*, 2nd Edition, Addison Wesley, 2007.

مراجع کمکی

- D. Grune, H. Bal, C. Jacobs, K. Langendoen, *Modern Compiler Design*, John Wiley, 2001.
- J. Trembly, P. Sorenson, *Theory and Practice of Compiler Writing*, McGraw Hill, 1985.
- C. Fisher, R. LeBlanc, *Crafting a Compiler with C*, Benjamin Cummings, 1991.



طراحی پایگاه داده‌ها (۴۰۳۸۴)

Database Design

طراح درس: سید محمد تقی روحانی رانکوهی

مقطع: کارشناسی	گرایش: نرم افزار
نوع درس: اصلی	واحد: ۳
هم‌نیاز: تحلیل و طراحی سیستم‌ها	پیش‌نیاز: ساختمان داده‌ها و الگوریتم‌ها

کلیات

در این درس با مفاهیم طراحی پایگاه داده‌ها آشنا می‌شویم و در انتهای ترم انتظار می‌رود که دانشجو با مفاهیمی که در ریزموارد به آنها اشاره شده است، آشنایی کامل پیدا کرده باشد.

ریز مواد

- کلیات
- تعریف پایگاه داده؛ مشی فایلینگ و مشی پایگاهی؛ عناصر محیط پایگاه داده؛ انواع معماری سیستم پایگاهی (متمرکز، مشتری-خدمتگذار، توزیع شده).
- مدل سازی معنایی داده‌ها با روش ER و (EER)
- موجودیت؛ صفت؛ ارتباط؛ نمودار (E)ER، انواع دام‌ها؛ تکنیک‌های تخصیص، تعمیم، تجزیه، ترکیب و جمع‌بندی؛ ویژگی‌های روش مدل سازی معنایی؛ (مدل سازی داده‌ها با UML: مطالعه توسط دانشجو).
- ساختارهای داده‌ای
- آشنایی با ساختارهای داده‌ای جدولی (رابطه‌ای)، سلسله‌مراتبی، و شبکه‌ای؛ ضوابط مطالعه تطبیقی DSها؛ دلایل لزوم ساختار داده‌ای، پایگاه داده جدولی، زبان پایگاه داده جدولی (SQL).
- معماری سه سطحی پایگاه (پیشنهادی ANSI)
- دید (نمای ادراکی؛ دید داخلی و دید خارجی؛ تبدیلات بین سطوح؛ عملیات از دید خارجی در پایگاه داده‌ها و مشکلات مربوطه؛ کاربر؛ زبان میزبان؛ زبان داده‌ای فرعی.
- DBMS (وظایف؛ رده‌بندی؛ واحدهای تشکیل دهنده) - DBA (اعضاء تیم؛ جزئیات وظایف تیم DBA)
- ریزفعالیت‌های برای ایجاد سیستم پایگاهی؛ مزایا و معایب تکنولوژی پایگاه داده‌ها؛ استقلال داده‌ای فیزیکی و منطقی؛ محورهای مقایسه DBMSها؛ شرایط استفاده از تکنولوژی پایگاه و ..
- مفاهیم اساسی مدل داده‌ای رابطه‌ای
- رابطه و مفاهیم مربوطه؛ تفاوت‌های رابطه با جدول؛ میدان (دامنه)؛ انواع رابطه؛ رابطه‌های نرمال و غیر نرمال (مزایا و معایب هر یک) ؛ کلید در مدل رابطه‌ای.
- اصول طراحی پایگاه داده‌های رابطه‌ای
- روش بالا به پایین (تبدیل مدل سازی معنایی به طراحی منطقی)؛ (اصول کلی طراحی پایگاه داده‌های شیء‌گرا: مطالعه دانشجو).
- جامعیت در مدل رابطه‌ای
- قواعد کاربری؛ مکانیزم‌های اعمال قواعد جامعیت کاربری؛ قواعد C1 و C2؛ مولفه‌های مدل داده‌ای در مدل رابطه‌ای؛ مزایا و معایب مدل رابطه‌ای.
- عملیات در پایگاه رابطه‌ای
- جبر رابطه‌ای و حساب رابطه‌ای.

۱۰. اصول طراحی پایگاه داده رابطه‌ای (نرمال ترسازى رابطه‌ها: روش سنتز)
مفاهیمی از تئوری وابستگی؛ شرح صورت‌های نرمال؛ تجطیه مطلوب؛ مزایا و معایب نرمال ترسازى.

توجه: از مرحله معرفی ساختارهای داده‌ای جدولی و پس از آن، هر جا که لازم باشد، از SQL برای ارائه مثال استفاده می‌شود. دانشجو بایستی از هفته سوم و همروند با پیشرفت درس، SQL را مطالعه کند.

آزمون - تمرین - پروژه

- آزمون کتبی
- تمرین
- مطالعه بیشتر
- تحقیق و ارائه شفاهی آن
- پروژه

مراجع اصلی

- T. Connolly, and C., Begg, *Database Systems*, 3rd Edition 2002, 4th Edition 2005.
- C.J. Date, *An Introduction to Database Systems*, 8th Edition 2003, 7th Edition 2000, 6th Edition 1995.
- R. Elmasri, S. Navate, *Fundamental of Database Systems*, 4th Edition 2003, or 5th Edition 2007.
- R. Ramakrishnan, J. Gehrke, *Database Management Systems*, 2nd Edition, McGraw-Hill, 2000.
- A. Silberschartz, H.F. Korth, S. Sudarshan, *Database System Concepts*, 5th Edition, McGraw Hill, 2006.
- J. D.Ullman, J. Widom, *A First Course in Database Systems*, 2nd Edition, Prentice Hall, 2002.
- روحانی رانکوهی، مفاهیم بنیادی پایگاه داده‌ها، چاپ هشتم ۱۳۸۸، چاپ نهم ۱۳۸۹، چاپ دهم ۱۳۸۹.
- روحانی رانکوهی، مقدمه‌ای بر پایگاه داده‌ها، چاپ چهارم، ۱۳۸۷.
- روحانی رانکوهی، سیستم مدیریت پایگاه داده‌ها (مفاهیم و تکنیک‌ها)، چاپ سوم، ۱۳۸۶. (به منظور مطالعه بیشتر)
- --، پایگاه داده‌های شیء‌گرا و شیء-رابطه‌ای، ترجمه: روحانی رانکوهی، چاپ اول، ۱۳۸۷.
- روحانی رانکوهی، پرسش‌هایی در پایگاه داده‌ها (۴ گزینه‌ای)، چاپ اول، ۱۳۸۸.

مراجع فرعی (منابع مهندسی فایل‌ها)

- B. Salzberg, *File Structures: An Analytic Approach*, Prentice-Hall, 1988.
- G. Wiederhold, *File Organization for Database Design*, McGraw-Hill, 1987.
- روحانی رانکوهی، سیستم . ساختار فایل‌ها (مهندسی فایل‌ها)، چاپ بیست و چهارم ۱۳۸۸، چاپ بیست و پنجم ۱۳۸۹، چاپ بیست و ششم ۱۳۸۹، چاپ بیست و هفتم ۱۳۸۹.



تحلیل و طراحی سیستم‌ها (۴۰۴۱۸) Systems Analysis and Design

دانشکده مهندسی کامپیوتر

طراح درس: جعفر حبیبی

مقطع: کارشناسی	گرایش: نرم‌افزار
نوع درس: اصلی	واحد: ۳
هم‌نیاز: طراحی پایگاه داده‌ها	پیش‌نیاز: ارائه مطالب علمی و فنی، ساختمان داده‌ها و الگوریتم‌ها

کلیات

هدف از این درس آشنایی دانشجویان کارشناسی نرم‌افزار با مفاهیم تحلیل و طراحی سیستم‌های نرم‌افزاری است. در این درس دانشجویان با انواع سیستم‌های اطلاعاتی، چرخه حیات تولید و توسعه نرم‌افزار، روش‌های مختلف ایجاد نرم‌افزار، تحلیل و طراحی ساخت‌یافته و مفاهیم مدیریت پروژه آشنا می‌شوند.

ریز مواد

۱. مفاهیم کلی توسعه سیستم
 - a. آشنایی با سیستم‌های اطلاعاتی، دینفعان سیستم، محرکان مختلف سیستم‌های اطلاعاتی، فرایند ایجاد نرم‌افزار (۱ جلسه)
 - b. انواع و اجزای سیستم‌های اطلاعاتی، مفاهیم دانش، فرایند و ارتباطات (۱ جلسه)
 - c. مدل بلوغ-قابلیت، چرخه حیات توسعه سیستم و روش‌های مختلف ایجاد سیستم، ابزارهای خودکار و تکنولوژی‌های مربوطه (۱ جلسه)
 - d. مدیریت پروژه، چرخه مدیریت پروژه، ابزارهای مدیریت پروژه (۲ جلسه)
۲. تحلیل سیستم
 - a. روش‌های مختلف تحلیل سیستم، طرح مراحل تحلیل سیستم، فعالیت‌های هر مرحله، ورودی و خروجی‌های آن‌ها (۱ جلسه)
 - b. جمع‌آوری و تحلیل نیازمندی‌ها، فنون کشف واقعیات و نیازمندی‌ها، مدل‌سازی موارد کاربردی (۲ جلسه)
 - c. مدل‌سازی و تحلیل داده، فرایند مدل‌سازی منطقی داده، ایجاد مدل موجودیت-رابطه و تحلیل آن (۱ جلسه)
 - d. مدل‌سازی فرایند، فرایند مدل‌سازی منطقی فرایندها، ایجاد مدل جریان داده و سنکرون‌سازی مدل‌های سیستم (۱ جلسه)
 - e. تحلیل امکان‌پذیری، تحلیل سود-هزینه و ارائه پروپوزال سیستم (۱ جلسه)
۳. طراحی سیستم
 - a. روش‌های مختلف طراحی سیستم، مراحل طراحی سیستم برای ساخت سیستم، مراحل طراحی سیستم برای یکپارچه‌سازی محصولات تجاری (۱ جلسه)
 - b. مدل‌سازی و معماری برنامه، نمودارهای جریان داده فیزیکی، معماری فناوری اطلاعات، استراتژی‌های معماری برنامه برای طراحی سیستم، مدل‌سازی معماری برنامه برای سیستم اطلاعاتی (۱ جلسه)
 - c. طراحی پایگاه داده، مقایسه سیستم‌های سنتی قابل و سیستم پایگاه داده، نرمال‌سازی، مفاهیم و طراحی پایگاه داده (۱ جلسه)
۴. ساخت و نگهداری سیستم
 - a. ساخت و پیاده‌سازی سیستم، عملکرد و پشتیبانی سیستم، بهبود سیستم (۱ جلسه)

آزمون - تمرین - پروژه

- آزمون: آزمون‌های میان‌ترم (۵۰٪ کل نمره)

- تمرین: چهار یا پنج تمرین که در طی آنها مستندات مراحل مختلف تحلیل و طراحی سیستم تحویل داده می‌شوند. این تمرینات در قالب پروژه نهایی درس تعریف می‌شوند. تمرینات در طول نیمسال تحویل داده می‌شوند. (۲۵٪ کل نمره)
- پروژه عملی: در ادامه تمرینات تحویلی درس، پروژه تعریف شده پیاده‌سازی می‌شود و سیستم اطلاعاتی توسعه داده شده در انتها تحویل داده می‌شود. (۲۵٪ کل نمره)

مرجع اصلی

- L. D. Bentley, J. L. Whitten, *Systems Analysis and Design for the Global Enterprise*, 7th Edition, McGraw-Hill, 2007.

مراجع کمکی

- J. A., Hoffer J. F. George, J. S. Valacich, *Modern Systems Analysis and Design*, 4th Edition, Prentice Hall, 2004.
- J. L. Whitten, L. D. Bentley, *Systems Analysis and Design Methods*, 7th Edition, McGraw Hill, 2005.
- J. Whitten, L. Bentley, *Introduction to Systems Analysis & Design*, McGraw Hill, 2006.
- K. E. Kendall, J. E. Kendall, *Systems Analysis and Design*, 8th Edition, Prentice Hall, 2010.
- B. H. Dennis, R. M. Wixom, *Systems Analysis and Design*, 4th Edition, John Wiley, 2008.
- S. Wasson, *Systems Analysis, Design, And Development: Concepts, Principles, and Practices*, John Wiley, 2006.
- G. B. Shelly, H. J. Rosenblatt, *Systems Analysis and Design*, 8th Edition, Shelly Cashman Series, 2010.



طراحی زبان‌های برنامه‌سازی (۴۰۳۶۴) Design of Programming Languages

طراح درس: محمد ایزدی

مقطع: کارشناسی	گرایش: نرم‌افزار
نوع درس: اصلی	واحد: ۳
هم‌نیاز: -	پیش‌نیاز: طراحی کامپایلرها

کلیات

هدف از ارایه این درس آشنایی دانشجویان با اصول و مسایل مرتبط با طراحی زبان‌های برنامه‌سازی و همچنین روش‌ها و ساختارهای داده‌ای به کار رفته در پیاده‌سازی یا محقق کردن محیط برنامه نویسی و اجرای برنامه‌ها با زبان‌های برنامه‌سازی در سنت‌های موجود است. همچنین یکی از اهداف این درس آشنایی دانشجویان با سیر تحولات مفاهیم و روش‌های طراحی و پیاده‌سازی نسل‌های مختلف زبان‌های برنامه‌سازی از دیدگاه تاریخی و آشنایی با زبان‌های اصلی که موجب ایجاد و یا تحولات اساسی در این سنت‌های برنامه‌سازی شده‌اند می‌باشد. به طور کلی سنت‌های برنامه‌سازی یا روش‌های تعریف مفهوم برنامه که بر اساس آنها زبان‌های برنامه‌سازی مناسب طراحی می‌شوند را می‌توان به چهار دسته تقسیم کرد: سنت برنامه‌سازی امری-رویه‌ای (Imperative-Procedural)، سنت برنامه‌سازی شی‌گرا (Object Oriented)، سنت برنامه‌سازی منطقی یا مبتنی بر قاعده (Rule Based) و سنت برنامه‌سازی تابعی (Functional). با توجه به اهداف بالا و نظر به آن که دانشجویانی که این درس را اخذ می‌کنند قبلاً با برنامه‌سازی به روش‌های امری-رویه‌ای، شی‌گرا و مبتنی بر قاعده آشنایی نسبی پیدا کرده‌اند ارایه این درس به دو نیم‌ترم تقسیم می‌شود. در نیم‌ترم اول مروری بر مفاهیم اصلی در طراحی زبانها و برخی زبانهای مهم ارایه خواهند شد. منبع اصلی این بخش کتاب MacLennan خواهد بود. نیم‌ترم دوم به ارایه تجربی و گام به گام مراحل طراحی یک زبان برنامه‌سازی و پیاده سازی مفسر آن با استفاده از یک زبان برنامه نویسی تابعی (مانند زبان Scheme) اختصاص دارد. در نتیجه نه تنها به صورت نظری که در عمل نیز دانشجویان با طراحی و پیاده‌سازی زبانهای برنامه‌سازی آشنا می‌شوند. منبع اصلی این بخش کتاب Friedman خواهد بود. ضمناً در ارایه این درس فرض بر آن است که دانشجویان با یکی از زبان‌های برنامه‌سازی ++C و Java به طور کامل و با زبان‌های C و Prolog آشنایی اجمالی دارند.

ریز مواد

۱. مقدمه: معرفی شامل تاریخچه تکاملی زبانهای برنامه‌سازی، معرفی سنتهای برنامه‌سازی و مسایل اصلی در پیاده‌سازی زبانها (۲ جلسه).
۲. معرفی تکاملی سیر تحول مفاهیم اصلی برنامه‌سازی در نسل‌های متکامل شده زبان‌ها به ترتیب شامل زبان‌های Fortran (۱ جلسه)، Algol (۲ جلسه)، Pascal (۱ جلسه)، ADA (۲ جلسه)، Common LISP (۳ جلسه)، Smalltalk (۱ جلسه)، Prolog (۱ جلسه). در صورت آشنایی قبلی دانشجویان با مفاهیم اصلی در برنامه نویسی شی‌گرا یا برنامه نویسی مبتنی بر قاعده دو مورد آخر می‌تواند به ارایه دانشجویی واگذار شود.
۳. مروری بر روش‌های پیاده‌سازی و کنترل روند اجرای برنامه‌ها. پیاده‌سازی ساختار بلوکی، کنترل فراخوانی و اجرای زیربرنامه‌ها و مدیریت حافظه (۲ تا ۳ جلسه).
۴. معرفی اجمالی برخی زبان‌های برنامه‌سازی امروزی‌تر مانند ML، Python و Haskell (۲ تا ۳ جلسه). می‌تواند توسط دانشجویان علاقه‌مند ارایه شود.
۵. طراحی و پیاده‌سازی تجربی و گام به گام یک زبان برنامه‌سازی با استفاده از زبان برنامه نویسی تابعی (مانند Scheme) که شامل فصل‌های ۱ تا ۶ کتاب Friedman خواهد بود (۱۲ تا ۱۴ جلسه همراه با سنجش میزان گیرایی مطالب برای دانشجویان).

آزمون - تمرین - گزارش پژوهشی

- آزمون‌های کوتاه در طول ترم (۴۰٪ کل نمره).
- آزمون پایان ترم (۳۰٪ کل نمره).
- تمرین: چند سری تمرین جهت آشنایی با برنامه‌های تابعی (۱۵٪ کل نمره + ۲ نمره اضافی).
- گزارش پژوهشی: ارائه‌ای از یکی از زبان‌های برنامه‌سازی یا زبان‌های توصیف نرم‌افزار که در کلاس مورد بحث قرار نمی‌گیرند (۱۵٪ کل نمره + ۲ نمره اضافی).

مراجع اصلی

- D. P. Friedman, M. Wand, *Essentials of Programming Languages*, 3rd Edition, MIT Press, 2008.
- B. J. MacLennan, *Principles of Programming Languages Design, Evaluation, and Implementation*, 3rd Edition, Oxford University Press, 1999.
- T. Pratt, M. Zelkowitz, *Programming Languages: Design and Implementation*, 4th Edition, Prentice-Hall, 2000.



مهندسی نرم افزار (۴۰۴۷۴) Software Engineering

طراح درس: سید حسن میریان حسین آبادی

مقطع: کارشناسی	گرایش: نرم افزار
نوع درس: اصلی	واحد: ۳
هم نیاز: -	پیش نیاز: تحلیل و طراحی سیستم‌ها

کلیات

هدف از این درس پرداختن به نکات مهندسی است که در کلیه مراحل تولید نرم افزار باید رعایت گردد. دانشجویان با مباحث ساخت نرم افزار (برنامه سازی)، تجزیه و تحلیل نیازمندی‌ها و طراحی نرم افزار در درس‌های قبلی آشنا شده‌اند. در این درس هدف آموزش روش جدیدی برای تجزیه و تحلیل نیازها و یا طراحی نرم افزار نیست. بلکه هدف آموزش تولید نرم افزار به صورت یک محصول مهندسی است، مانند سایر محصولاتی که در سایر رشته‌های مهندسی تولید می‌گردد. در این درس ابتدا تفاوت محصولی که به روش مهندسی تولید می‌گردد با محصولی که به روش هنری تولید می‌شود بیان می‌شود. سپس انتظاراتی که یک محصول مهندسی باید برآورده سازد تشریح می‌گردد. در ادامه درس با تاکید بر روش‌های مهندسی تولید از جمله مدل سازی، قابل اندازه گیری و ارزیابی بودن، درستی یابی و اعتبارسنجی محصولات بینابینی، مروری بر دست‌آوردهای علمی در این زمینه در کلیه مراحل تولید نرم افزار انجام می‌شود. با توجه به اینکه در درس‌های قبلی دانشجویان با مباحث توصیف صوری نیازها، اندازه گیری، تخمین و آزمون کمتر آشنا شده‌اند در این درس این فصول مورد تاکید بیشتر قرار می‌گیرد. در انتها فعالیت‌های حمایتی از جمله مدیریت پروژه، زمانبندی، مدیریت ریسک، مدیریت پیکربندی و تضمین کیفیت با تاکید بر تاثیر آن‌ها در تولید نرم افزار به صورت مهندسی مرور می‌شود.

ریز مواد

۱. مقدمه (۲ جلسه)
۲. فرایند-مدل (۲ جلسه)
۳. تولید چابک (۱ جلسه)
۴. درک نیازها (۱ جلسه)
۵. روش‌های صوری (۵ جلسه)
۶. مفاهیم طراحی (۱ جلسه)
۷. طراحی معماری (۱ جلسه)
۸. طراحی واسط (۱ جلسه)
۹. طراحی مبتنی بر الگو (۱ جلسه)
۱۰. استراتژی‌های آزمون (۱ جلسه)
۱۱. روش‌های آزمون (۴ جلسه)
۱۲. اندازه گیری محصول (۱ جلسه)
۱۳. اندازه گیری فرایند و پروژه (۱ جلسه)
۱۴. برآورد (۱ جلسه)
۱۵. مفاهیم کیفیت (۱ جلسه)
۱۶. روش‌های مرور (۱ جلسه)
۱۷. تضمین کیفیت (۱ جلسه)
۱۸. مدیریت پیکربندی (۱ جلسه)
۱۹. مدیریت پروژه (۱ جلسه)

۲۰. زمانبندی (۱ جلسه)

۲۱. مدیریت ریسک (۱ جلسه)

آزمون - تمرین - سمینار

- ۳ تمرین نظری-عملی در طول ترم (۲۰٪ کل نمره)
- ۳ آزمون تستی از مطالب درس در طول ترم (۳۰٪ کل نمره)
- حدود ۵ آزمون کوچک در طول ترم (۱۰٪ کل نمره) (نمره اضافه)
- آزمون نهایی تشریحی و تستی (۵۰٪ کل نمره)
- دانشجویان به طور اختیاری سمیناری را از فصولی از کتاب که تدریس نمی‌شود و موضوعات مرتبط با مطالب درس پس از گرفتن تایید ارایه می‌نمایند (۱۰٪ کل نمره (نمره اضافه)).

مرجع اصلی

- R.S. Pressman, *Software Engineering: A Practitioner's Approach*, 7th Edition, McGraw-Hill, 2010.
- P. Ammann, J. Offutt, *Introduction to Software Testing*, Cambridge University Press, 2008.
- J. Woodcock, J. Davies, *Using Z: Specification, Refinement, and Proof*, Prentice Hall, 1996.



آزمایشگاه مهندسی نرم افزار (۴۰۴۰۴) Software Engineering Laboratory

طراح درس: سید حسن میریان

مقطع: کارشناسی	گرایش: نرم افزار
نوع درس: اصلی	واحد: -
هم نیاز: -	پیش نیاز: مهندسی نرم افزار

کلیات

این درس برای دانشجویان کارشناسی ارائه می شود و هدف از آن پرداختن به مباحث کاربردی مهندسی نرم افزار است. در این درس روش های مهندسی در قالب پروژه های عملی تجربه می شوند. کار آزمایشگاه در پنج حوزه اصلی مهندسی نرم افزار شامل مهندسی نیازمندی ها، تحلیل، طراحی، پیاده سازی و آزمون انجام می پذیرد. آزمایشگاه از ۱۰ جلسه سه ساعته تشکیل می شود. به هر گروه یک پروژه تخصیص داده می شود که کار خود را تا پایان ترم بروی آن پروژه و مستندات آن انجام خواهد داد.

سرفصل های کلی جلسات ده گانه آزمایشگاه

۱. معرفی درس، گروه بندی، تعیین پروژه و سایر مباحث اولیه درس
۲. مهندسی نیازمندی ها
۳. تحلیل (قسمت ۱)
 - معرفی کلی بحث تحلیل و جایگاه آن نسبت به دو فعالیت مهندسی نیازمندی ها و طراحی
 - پرداختن به چستی به جای چگونگی
 - نمودار فعالیت سطح بالا مربوط واقعیت بخشی به موارد کاربرد
 - نحوه شناسایی کلاسهای تحلیل و نمودار کلاسها
 - الگوهای تحلیل (در صورت امکان، فعالیت اضافه)
۴. تحلیل (قسمت ۲)
 - نمودار ترتیب و استفاده از آن در تحلیل
 - Package Diagram
 - الگوهای تحلیل (در صورت امکان، فعالیت اضافه)
۵. طراحی (قسمت ۱)
 - معرفی کلی بحث طراحی
 - الگوها و معیارهای GRASP: کتاب Larman فصل ۱۷ و ۲۵ + ارائه کامل مثال ارائه شده در فصل ۱۷,۸ این کتاب با جزئیات آن
۶. طراحی (قسمت ۲)
 - انواع Cohesion و Coupling با ذکر مثال
 - نمودار کلاسها با همه جزئیات آن
 - منبع: پوشش کامل از مطالب فصلهای ۳ و ۵ از کتاب UML Distilled ویرایش سوم
۷. پیاده سازی: Refactoring
 - معرفی بحث Refactoring:
 - منبع: کتاب Refactoring نوشته Martin Fowler

- ارائه یک مثال از کد پیاده‌سازی شده و Refactoring در آن
- منبع: پوشش کامل فصل ۱ کتاب Refactoring نوشته Martin Fowler
- معرفی بوهای بد در کد (Bad Smells)
- منبع: فصل ۳ کتاب Refactoring نوشته Martin Fowler
- ۸. آزمون (قسمت ۱): Unit Testing
- مفاهیم Unit Testing
- معرفی کلی چارچوبهای موجود در این زمینه برای زبان‌ها و محیط‌های برنامه‌سازی مختلف
- معرفی کامل JUnit و ابزار جانبی مرتبط با آن و پشتیبانی‌های IDEها از آن
- ارائه یک مثال از نحوه استفاده از JUnit و اجرای آن
- ۹. آزمون (قسمت ۲): ISP and PPC Testing Techniques
- Input Space Partitioning
- Graph Based Prime Path Coverage (Based on Source Code)
- ۱۰. جلسه پایانی و جمع‌بندی
- ارائه‌های اختیاری
- ابزارهای مدیریت پیکربندی نرم افزار (Software Configuration Management Tools)
- ابزارهای ارزیابی پوشش آزمون به همراه ارائه یک مثال عملی (Test Coverage Tools)

نحوه اداره آزمایشگاه

- آزمایشگاه از ۱۰ جلسه ۳ ساعته تشکیل می‌شود.
- کار آزمایشگاه در پنج حوزه اصلی مهندسی نرم‌افزار شامل مهندسی نیازمندی‌ها، تحلیل، طراحی، پیاده‌سازی و آزمون انجام می‌پذیرد.
- برنامه هر جلسه آزمایشگاه:
- ۱. ارزیابی فردی کار گروه‌های دیگر: ۲۰ دقیقه
- ۲. ارائه کار انجام شده گروه‌ها و دفاع از طرح خود: در مجموع ۴۰ دقیقه
- ۳. ارائه مفاهیم و ابزار (مربوط به موضوع آن جلسه) توسط گروه‌ها: ۴۰ دقیقه
- ۴. تکمیل مباحث مفاهیم و ابزار در صورت نیاز توسط TA + استراحت: ۲۰ دقیقه
- ۵. انجام کار گروهی در کلاس: مرور، بازبینی و اصلاح مدل‌ها و کارهای قبلی خود: ۴۰ دقیقه
- ۶. ارائه پیش‌نویس مدل‌های اصلاح شده و نتایج کار گروهی در آزمایشگاه به TA
- ۷. تهیه گزارش کار آزمایشگاه تا هفته بعد توسط گروه‌ها و ارسال آن به اعضای کلاس/آوردن در کلاس

جزئیات کار جلسات آزمایشگاه

- ارائه‌ها:
- ۱. با توجه به پنج حوزه اصلی مهندسی نرم‌افزار پوشش داده شده در آزمایشگاه، هر هفته یک گروه می‌بایست مفاهیم و ابزار مرتبط با آن مبحث را در مدت حدود ۴۰ دقیقه ارائه نماید.
- ۲. گروه‌های دو و سه نفره، یک ارائه خواهند داشت. گروه‌های چهار نفره ممکن است دو بار ارائه داشته باشند که این موضوع براساس نظر TA تعیین می‌شود.
- ۳. ارائه‌های هر گروه توسط TA ارزیابی شده و نمره آن مربوط به کل گروه ارائه دهنده خواهد بود.
- ۴. در برخی مباحث که دانشجویان ممکن است آشنایی کمتری با مباحث مورد نظر داشت باشند، مانند موضوع آزمون نرم‌افزار، بخشی از ارائه توسط TA انجام می‌پذیرد.
- ۵. مباحثی که در هر ارائه می‌بایست پوشش داده شود عبارتند از:

- a. بیان مفاهیم موضوع مورد ارائه از نظر تئوری
- b. نمودارهای مرتبط UML و مباحث مربوط به آن
- c. الگوها و نمونه‌های موفق^۲ مرتبط با موضوع (مثلاً الگوهای تحلیل، طراحی، معماری و ...)
- d. معیارهای اندازه‌گیری و ارزیابی کارها و چگونگی تشخیص کار قوی و ضعیف
- e. بوهای بد^۳ و پادالگوها^۴
- f. ابزار مرتبط برای انجام کار، قابلیت‌ها و چگونگی عملکرد
- مثلاً ابزار مدل‌سازی، ابزار تولید کد، ابزار آزمون، ...
 - ابزار معرفی شده می‌بایست برای دانشجویان قابل دسترسی و استفاده باشد.
۶. اسلایدها و نام و لینک منابع مرتبط، می‌بایست توسط گروه ارائه دهنده، تا ۲۴ ساعت بعد از ارائه برای همه اعضای کلاس به صورت ایمیل ارسال گردد.
۷. اسلایدها، منابع و ابزار ارائه شده توسط هر گروه، می‌بایست در قالب CD در همان جلسه ارائه، به TA درس داده شود.
- **چگونگی انجام کار گروهی آزمایشگاه**
 ۱. نحوه گروه بندی و اختصاص پروژه‌ها
 - a. حتی‌الامکان کلاس به گروه‌های حداکثر ۴ نفره تقسیم می‌شود.
 - b. همه افراد، پروژه‌های درس تحلیل و طراحی (SAD) یا طراحی شی‌گرا (OOD) با همه جزئیات و مستندات و پیاده سازی را ارسال می‌کنند.
 - c. پروژه‌هایی که برای انجام کار گروهی مورد نظر مناسب باشند، توسط TA انتخاب شده و به گروه‌ها تخصیص داده می‌شود.
 - d. به هر گروه یک پروژه (حتی المقدور پروژه یکی از اعضای همان گروه) تخصیص داده می‌شود که کار خود را تا پایان ترم بر روی آن پروژه و مستندات آن انجام خواهند داد.
 - e. سعی می‌شود در مجموع، دو صورت مسئله (با پیاده‌سازی‌های متفاوت) به عنوان پروژه‌های گروه‌ها انتخاب شود تا از پراکندگی تعاریف مسئله جلوگیری شود.
 - f. تخصیص پروژه‌ها حداکثر تا پایان جلسه اول کلاس انجام می‌پذیرد.
 ۲. کار گروهی در کلاس
 - a. هر جلسه کلاس، به یک موضوع از مباحث مهندسی نرم‌افزار اختصاص دارد که ارائه نیز در همان موضوع انجام شده است.
 - b. کار گروهی هر جلسه آزمایشگاه، مرور کار انجام شده قبلی گروه (از مستندات پروژه تخصیص داده شده به هر گروه)، ارزیابی نقاط قوت و ضعف آن و بازبینی و اصلاح آن می‌باشد.
 - c. به عنوان مثال، در جلسه مربوط به طراحی، مستندات طراحی قبلی مرور شده و با توجه به دانش دانشجویان و ارائه انجام شده و معیارهای ارزیابی و ...، نقاط قوت و ضعف آن طراحی استخراج شده و در تعامل گروهی، آن را اصلاح می‌نمایند و مدل طراحی جدیدی توسط گروه ارائه می‌شود.
 - d. این کار بین ۳۰ دقیقه تا یک ساعت می‌تواند به طول بیانجامد.
 - e. در زمان انجام کار گروهی، هر گروه می‌تواند یک notebook داشته باشد و از آن استفاده نماید.
 - f. پس از پایان کار گروهی یا پایان ساعت آزمایشگاه، پیش‌نویسی از کار گروهی انجام شده حاوی نتیجه کار (مدل جدید)، عناوین معیارها و الگوهای مورد استفاده و نقاط قوت مدل جدید نسبت به قبلی (فقط عناوین آن) به TA کلاس ارائه می‌گردد.
 - g. این پیش‌نویس در پایان کلاس می‌بایست به TA ارائه گردد که می‌تواند به صورت فایل الکترونیکی و یا کاغذ دستنویس باشد.

¹Pattern

²Best Practice

³Bad Smells

⁴Anti-Pattern

h. در صورت نیاز، TA می‌تواند یک کپی از این کاغذ پیش‌نویس را در اختیار تیم قرار دهد (برای انجام گزارش کار آزمایشگاه)

i. نتیجه کار کلاس و پیش‌نویس تهیه شده، توسط TA ارزیابی خواهد شد و بخش مهمی از نمره آن کلاس را تشکیل می‌دهد.

• تهیه گزارش کار آزمایشگاه

۱. هر گروه تا هفته آینده، فرصت دارد کار گروهی انجام شده در کلاس خود را تکمیل نماید و نواقص احتمالی آن را برطرف کرده و نتیجه‌نهایی کار خود را در قالب یک گزارش کار آزمایشگاه ارائه دهد.

۲. گزارش کار آزمایشگاه می‌بایست مطالب زیر را شامل شود:

- مدل قبلی (کار اولیه مطابق مستندات پروژه تخصیص داده شده به هر گروه)
- مدل (کار) جدید (پس از بازنگری)
- روش انجام کار: معیارها، الگوهای مورد استفاده، روش کار گروهی، ابزار مورد استفاده
- نقاط ضعف مدل قبلی
- نقاط قوت مدل جدید
- نقاط ضعف احتمالی مدل جدید (در tradeoff)
- راه‌حل‌های جایگزین (احتمالی)

۳. گزارشات کار می‌بایست در قالب گزارش علمی باشد، فصل‌بندی مناسب داشته باشد، به زبان فارسی باشد، از فونت ۱۲ و حاشیه 2x2x2x2 در کاغذ A4 برای تهیه آن استفاده شده باشد.

۴. گزارش کار آزمایشگاه، توسط TA ارزیابی می‌شود.

۵. گزارش کار آزمایشگاه، توسط دانشجویان نیز در ابتدای جلسه بعد، ارزیابی می‌شود.

۶. از آنجا که گزارش کار هر گروه می‌بایست در جلسه بعد توسط اعضای گروه‌های دیگر ارزیابی شود و این ارزیابی فردی (تک نفره) خواهد بود، هر گروه می‌بایست گزارش کار آزمایشگاه خود را حداکثر تا ساعت ۲۴ دو روز قبل از برگزاری جلسه بعد، به همه اعضای کلاس خود و TA به صورت ایمیل ارسال نماید. موارد استثناء از طرف TA اعلام خواهد شد.

- به عنوان مثال، دانشجویانی که چهارشنبه ظهر کلاس دارند، گزارش کار خود را می‌بایست تا دوشنبه شب ارسال نمایند.

۷. گروه‌هایی که به هر دلیل موفق به ارسال گزارش خود تا زمان مقرر نشوند، می‌بایست ۸ عدد کپی از گزارش کار کامل خود را به صورت پرینت‌شده به کلاس بیاورند. در غیر این صورت، نمره گزارش کار را از دست خواهند داد.

• فعالیت ارزیابی کار سایر گروه‌ها در کلاس

۱. در ۲۰ دقیقه ابتدای هر جلسه، هر فرد می‌بایست گزارش کار تهیه شده توسط دو گروه دیگر (غیر از گروه خودش) را ارزیابی نماید.

۲. این ارزیابی براساس معیارهایی که دانشجو در جلسه قبل، انجام کار گروهی و طول هفته گذشته آموخته است، انجام می‌پذیرد.

۳. اینکه هر فرد، کار کدام گروه‌ها را می‌بایست ارزیابی نماید، حداکثر تا ۴۸ قبل از تشکیل کلاس، از طریق ایمیل، به اطلاع هر فرد خواهد رسید.

۴. هر فرد در صورت تمایل، می‌تواند قبل از تشکیل کلاس، فعالیت ارزیابی خود را انجام داده و نتیجه آن را به کلاس بیاورد. البته این کار در صورتی امکان‌پذیر است که گروه‌های مربوطه، گزارش خود را در مهلت مقرر ارسال کرده باشند.

۵. ارزیابی تنها در ۲۰ دقیقه اول کلاس و براساس مستندات مکتوب (گزارش کار) انجام می‌پذیرد و پس از آن، نتیجه ارزیابی از کسی پذیرفته نخواهد شد. لذا از اعضای کلاس می‌بایست از ابتدای کلاس حضور داشته باشند.

۶. ارزیابی هر فرد، دانش وی در موضوع مورد ارزیابی را نشان می‌دهد. بنابراین ارزیابی هر فرد، توسط TA کلاس ارزیابی خواهد شد و بخشی از نمره آن جلسه را به خود اختصاص خواهد داد.

۷. نتایج ارزیابی‌ها، در نمره گزارش کار مورد ارزیابی نیز موثر خواهد بود.

۸. در زمان ارزیابی، همه اعضای کلاس در صورت تمایل می‌توانند از کامپیوتر شخصی خود استفاده نمایند.

۹. مشورت در زمان ارزیابی، مجاز می‌باشد.

۱۰. پس از ۲۰ دقیقه ابتدائی کلاس، هر گروه ۱۰ دقیقه فرصت خواهد داشت تا کار خود را به طور مختصر ارائه داده و از کار خود در برابر انتقادات مطرح شده توسط سایر دانشجویان، دفاع نماید.

- M. Fowler, *Analysis Patterns: Reusable Object Models*, Addison-Wesley, 1996.
- M. Fowler, *Refactoring: Improving the Design of Existing Code*, Addison-Wesley, 1999.
- M. Fowler, *UML Distilled*, 3rd Edition, Addison-Wesley, 2004.
- E. Gamma, R. Helm, R. Johnson, J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley, 1995.
- C. Larman, *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development*, 3rd Edition, Prentice-Hall, 2004.



طراحی شیء‌گرای سیستم‌ها (۴۰۴۸۴)

Object-Oriented Systems Design

طراح درس: رامان رامسین

مقطع: کارشناسی	گرایش: نرم‌افزار
نوع درس: اصلی	واحد: ۳
هم‌نیاز: -	پیش‌نیاز: تحلیل و طراحی سیستم‌ها

کلیات

هدف این درس، آشنا کردن دانشجویان کارشناسی نرم‌افزار با مفاهیم، اصول و روش‌های تحلیل و طراحی شیء‌گرای سیستم‌های نرم‌افزاری است. دانشجویان ضمن آشنایی کامل با یک متدولوژی مطرح تحلیل و طراحی شیء‌گرا (نسل سوم)، با الگوهای طراحی GoF و چگونگی بکارگیری آنها نیز آشنا خواهند شد.

ریز مواد

۱. مقدمه - مروری بر شیء‌گرایی و معرفی تاریخچه تکاملی تحلیل و طراحی شیء‌گرا (۱ جلسه - مدت هر جلسه، نود دقیقه است)
۲. معرفی اجمالی زبان مدلسازی یکپارچه (UML) (۱ جلسه)
۳. معرفی اجمالی فرایند یکپارچه ایجاد نرم‌افزار (متدولوژی USDP) و مقایسه با متدولوژی RUP (۱ جلسه)
۴. مراحل و جریانهای کاری در USDP
 - مراحل چهارگانه (۱ جلسه)
 - جریان کاری خواسته‌ها - شناسایی و مدل‌سازی موارد کاربرد (۲ جلسه)
 - جریان کاری تحلیل
 - شناسایی و مدل‌سازی اشیاء و کلاسهای تحلیل (۱ جلسه)
 - شناسایی و مدل‌سازی روابط بین اشیاء و کلاسهای تحلیل (۲ جلسه)
 - بسته‌های تحلیل (۱ جلسه)
 - محقق‌سازی موارد کاربرد در تحلیل (۲ جلسه)
 - مدل‌سازی فعالیتها (۱ جلسه)
 - جریان کاری طراحی
 - شناسایی و مدل‌سازی اشیاء و کلاسهای طراحی (۱ جلسه)
 - پالایش روابط (۱ جلسه)
 - واسطها و مؤلفه‌ها (۱ جلسه)
 - محقق‌سازی موارد کاربرد در طراحی (۲ جلسه)
 - مدل‌سازی حالتها (۱ جلسه)
 - جریان کاری پیاده‌سازی (۱ جلسه)
 - مستقرسازی (۱ جلسه)
۵. الگوهای طراحی
 - اصول و قواعد طراحی: اصول ششگانه پایه، الگوهای GRASP، طراحی بر اساس قرارداد (۱ جلسه)
 - معرفی الگوهای طراحی: مبانی و الگوهای پایه Coad (۱ جلسه)

• الگوهای طراحی GoF

- الگوهای آفرینشی: Singleton, Prototype, Builder, Abstract Factory, Factory Method (۲ جلسه)
- الگوهای ساختاری: Proxy, Flyweight, Facade, Decorator, Composite, Bridge, Adapter (۲ جلسه)
- الگوهای رفتاری: State, Observer, Memento, Mediator, Iterator, Command, Chain of Responsibility (۳ جلسه)
- Visitor, Strategy

آزمون - تمرین - پروژه

- آزمون: آزمون‌های میان‌ترم و پایان‌ترم (۶۰٪ کل نمره)
- تمرین و پروژه: تمرینات در قالب یک پروژه درسی تحلیل و طراحی، تعریف شده و بتدریج در طول نیمسال انجام و تحویل داده می‌شوند. (۴۰٪ کل نمره)

مراجع اصلی

- J. Arlow, I. Neustadt, *UML 2 and the Unified Process*, 2nd Edition, Addison Wesley, 2005.
- G. Booch, R. A. Maksimchuk, M. W. Engel, B. J. Young, J. Conallen, K. A. Houston, *Object-Oriented Analysis and Design with Applications*, 3rd Edition, Addison Wesley, 2007.
- E. Gamma, R. Helm, R. Johnson, J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison Wesley, 1995.
- C. Larman, *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development*, 3rd Edition, Prentice-Hall, 2004.



گرافیک کامپیوتری (۴۰۴۴۷)

Computer Graphics

طراح درس: منصور جمزاد

مقطع: کارشناسی	گرایش: نرم افزار
نوع درس: اختیاری	واحد: ۳
هم‌نیاز: -	پیش‌نیاز: -

کلیات

هدف این درس آشنا کردن دانشجویان با مفاهیم پایه ای گرافیک کامپیوتری است. تکیه اصلی درس بر روی گرافیک کامپیوتری سه بعدی ، مفاهیم نور پردازی ، تولید صحنه های گرافیکی و بازی های کامپیوتری با استفاده از نرم افزار OpenGL است. این نرم افزار در محیط زبان های برنامه نویسی سطح بالا مانند C, C++, Java قابل استفاده است. انتظار میرود که دانشجویان با یکی از این زبان ها آشنائی کامل داشته باشند و در طول ترم استفاده از OpenGL را نیز بیاموزند.

ریز مواد

- آشنائی با مفاهیم کلی شامل طراحی به کمک کامپیوتر ، هنر در کامپیوتر، سرگرمی، آموزش و یادگیری، مصور سازی، رابط کاربر در محیط های گرافیک (۲ جلسه)
- مروری بر سیستم های گرافیک شامل دستگاه های نمایش و مکانیزم تولید تصویر در آنها، دستگاه های ورودی و چاپگرها، نرم افزارهای گرافیک (۲ جلسه)
- روش های نمایش سه بعدی و نرم افزارهای مربوطه (۱ جلسه)
- نمایش سه بعدی شامل سطوح چند ضلعی، خطوط و سطوح انحنا دار، سطوح درجه سوم، اشیا حبابی شکل، انواع اسپلین ها، سطوح اسپلین، تبدیل بین انواع اسپلین ها، درخت های هشت تائی و BSP برای نمایش اشیا و اشکال، تولید اشکال فراکتالی با استفاده از هندسه فراکتالی (۴ جلسه)
- نمایش سه بعدی اشیا: گرامر اشکال، مدل سازی مبتنی بر فیزیک اشیا، مصور سازی مجموعه داده ها، مفاهیم Projection، حجم های قابل رویت، قیچی کردن، دیدن حجم های سه بعدی و تشخیص سطوح قابل رویت (۳ جلسه)
- روش های تشخیص سطوح قابل رویت : روش back face، روش depth buffer، روش A-buffer، روش درخت هشت تائی، روش درخت BSP، روش ray casting، روش های مثلث بندی و ... (۳ جلسه)
- روش های نور پردازی و رنگ آمیزی سطوح: منابع نور، نمایش میزان روشنائی نور، الگوهای هاف تن، روش های رنگ آمیزی چند وجهی، روش های Ray-casting، مدل نورپردازی Radiosity، اضافه کردن جزئیات به تصویر (۵ جلسه)
- مدل های رنگ و کاربرد های آن : ویژگی های نور، مفاهیم استاندارد و دیگرام کروماسیتی، مدل های رنگ (YIQ, RGB, HSV، و CMY) تبدیل بین مدل های رنگ، انتخاب رنگ و کاربردهای آن (۱ جلسه)
- پویا نمائی کامپیوتری: طراحی دنباله پویا نمائی، توابع عمومی پویا نمائی، سیستم های فریم های کلیدی، مشخص کردن حرکت، حرکت تناوبی (۲ جلسه)
- روش های ضبط حرکت Motion Capture در پویانمائی (۲ جلسه)
- مفاهیم اولیه ساخت یک بازی کامپیوتری و موتورهای بازی (۳ جلسه)

آزمون - تمرین

- آزمون میان ترم (۵ نمره)
- آزمون پایان ترم (۸ نمره)
- تمرین‌های برنامه‌نویسی با استفاده از OpenGL (۷ نمره)

مراجع اصلی

- D. Hearn, M. P. Baker, *Computer Graphics with OpenGL*, 3rd Edition, Prentice Hall, 2004.
- E. Angel, *OpenGL: A Primer*, Addison Wesley, 2002.



مقطع: کارشناسی	گرایش: نرم‌افزار
نوع درس: اختیاری	واحد: ۳
هم‌نیاز: -	پیش‌نیاز: برنامه‌سازی پیشرفته

کلیات

در این درس به بررسی الگوهای مختلف در طراحی یک نرم‌افزار وب پرداخته میشود. همچنین فریم ورک سیمفونی آموزش داده خواهد شد و مطالب بر اساس آن پیاده‌سازی میشوند.

ریز مواد

۱. مقدمه‌ای بر HTML/Javascript/CSS (۴ جلسه)
۲. مقدمه‌ای بر PHP/MySQL (۴ جلسه)
۳. معماری‌های وب
 - لایه‌بندی (Layering) (۲ جلسه)
 - مدیریت (Domain Logic) (۲ جلسه)
 - انطباق با پایگاه‌داده‌های رابطه‌ای (۴ جلسه)
 - صورت ظاهری وب (Web Presentation) (۱ جلسه)
 - همزمانی و مسائل آن (Concurrency) (۱ جلسه)
 - جلسه (Session State) (۱ جلسه)
 - استراتژیهای توزیع (Distribution Strategies) (۱ جلسه)
۴. سیمفونی
 - مقدمه و نصب (۱ جلسه)
 - مفاهیم اولیه در تولید صفحه وب (۱ جلسه)
 - لایه کنترل (۱ جلسه)
 - لایه نمایش (۱ جلسه)
 - لایه تصمیم و منطق (۱ جلسه)
 - سیستم Routing (۱ جلسه)
 - فرمها (۱ جلسه)
 - تکنولوژی AJAX (۱ جلسه)
 - کش کردن (۱ جلسه)
 - محلی سازی (۱ جلسه)
 - تولید خودکار کد (۱ جلسه)
 - تست کردن (۱ جلسه)

آزمون - تمرین - پروژه

- ۳ تمرین عملی (۵ نمره)
- پروژه‌ی نهایی (۶ نمره)
- میان ترم (۳ نمره)
- آزمون نهایی (۶ نمره)

مراجع اصلی

- M. Fowler, *Patterns of Enterprise Application Architecture*, 1st Edition, Addison Wesley, 2002.
- F. Potentier, F. Zaninotto, *The Definitive Guide to Symfony*, Apress, 2007.



اصول برنامه نویسی موازی و همروند

Principles of Concurrent and Parallel Programming

طراح درس: حمید بیگی

مقطع: کارشناسی	گرایش: نرم افزار
نوع درس: اختیاری	واحد: ۳
هم نیاز: -	پیش نیاز: سیستم های عامل

کلیات

هدف از این درس آشنایی دانشجویان با مفاهیم برنامه نویسی همروند و موازی در محیط های مختلف می باشد. در این درس نخست دانشجویان با اصول برنامه نویسی همروند و موازی و شیوه های همگام سازی فرایند ها و نخ ها آشنا خواهند شد و سپس با ابزار های مختلف برنامه نویسی در چند محیط مانند لینوکس و جاوا و کتابخانه MPI و برنامه نویسی در پردازنده های چند هسته ای مانند پردازنده های گرافیکی و پردازنده Cell نیز آشنا خواهند شد. این درس دارای چند پروژه برنامه نویسی است و نمره پروژه ها بخش عمده ای از نمره درس را شامل خواهد شد.

ریز مواد

۱. مقدمه (۱ جلسه)
۲. بخش نخست : مبانی برنامه نویسی موازی و همروند (۱۸ جلسه)
 - مقدمه ای بر نخ و فرایند در دو محیط جاوا و سیستم های عامل مبتنی بر POSIX (با کمک Pthread) (۳ جلسه)
 - نواحی بحرانی و راه حل های نرم افزاری و سخت افزاری آن (۱ جلسه)
 - ارتباط بین فرایندها شامل ارتباطات مسدودکننده و غیر مسدودکننده (ارتباط فرایندها به کمک حافظه مشترک، Mailbox، Pipe، RPC، RMI و غیره) (۱ جلسه)
 - ساختار های موجود برای همگام سازی فرایندها و نخ ها و همچنین ساختارهای لازم برای پشتیبانی از نواحی بحرانی شامل سمافور، مانیتور، انواع قفل ها (lock)، موانع (Barrier)، حافظه تراکنشی (۴ جلسه)
 - برنامه نویسی در رایانه های با حافظه مشترک و شیوه پایان بخشیدن به برنامه های موازی (۳ جلسه)
 - برنامه نویسی در رایانه های با حافظه توزیع شده و استفاده از کتابخانه MPI برای برنامه نویسی موازی و همروند و شیوه پایان بخشیدن به برنامه های موازی (۴ جلسه)
 - آشنایی با چند نمونه ساختمان داده موازی مانند صف موازی، پشته موازی و لیست های پیوندی موازی (۲ جلسه)
۳. بخش دوم : برنامه نویسی پردازنده های چند هسته ای (بررسی دو پردازنده از سه پردازنده Cell, GPU, Intel) (۱۰ جلسه)
 - معماری پردازنده های چند هسته ای (۱ جلسه)
 - برنامه نویسی پردازنده های گرافیکی (۴ جلسه)
 - معماری پردازنده های گرافیکی و بطور مشخص معماری پردازنده های Nvidia
 - معماری CUDA و برنامه نویسی با کمک این محیط
 - برنامه نویسی پردازنده های Cell (۴ جلسه)
 - معماری پردازنده های Cell
 - برنامه نویسی در پردازنده های cell و برنامه نویسی SPE, PPE
 - برنامه نویسی پردازنده های اینتل (۴ جلسه)
 - معماری پردازنده های اینتل
 - برنامه نویسی در پردازنده های اینتل و برنامه نویسی با کمک TBB

- اشکال زدایی و ارزیابی کارایی برنامه ها (Debugger & Profiler) (۱ جلسه)

۴. بخش سوم: مباحث تکمیلی (۱ جلسه)

- آشنایی با openMP
- آشنایی با openCL

مراجع اصلی

- M. Herlihy, N. Shavit, *The Art of Multiprocessor Programming*, Morgan Kaufmann, 2008.
- David B. Kirk, Wen-mei W. Hwu, *Programming Massively Parallel Processors: A Hands-on Approach*, Morgan Kaufmann, 2010.
- IBM, *Cell Broadband Engine Programming Tutorial*, 2006.
- James Reinders, *Intel Threading Building Blocks: Outfitting C++ for Multi-Core Processor Parallelism*, O'Reilly Media, 2007.
- P. Pacheco, *Parallel Programming with MPI*, Morgan Kaufmann, 1996.

مرجع کمکی

- B. Chapman, G. Jost, R. van der Pas, *Using OpenMP Portable Shared Memory Parallel Programming*, MIT Press, 2008.



مبانی الگوریتم های انفورماتیک زیستی

Fundamentals of Bioinformatics Algorithms

طراح درس: حمید بیگی

مقطع: کارشناسی	گرایش: نرم افزار
نوع درس: اختیاری	واحد: ۳
هم نیاز: -	پیش نیاز: طراحی الگوریتم ها

کلیات

هدف از این درس آشنایی دانشجویان با مبانی الگوریتم های انفورماتیک زیستی است. در این درس دانشجویان با دسته های کلی الگوریتم های انفورماتیک زیستی آشنا خواهند شد و این سبب تقویت دانش الگوریتمی دانشجویان می گردد. علاوه بر جنبه های الگوریتمی درس، دانشجویان با پایگاه داده های موجود در این زمینه نیز آشنا خواهند شد

ریز مواد

۱. مقدمه ای بر زیست شناسی مولکولی (۲ جلسه)
۲. درخت ها و آرایه های پسوندی و کاربرد های آنها در انفورماتیک زیستی (۴ جلسه)
۳. هم ترازوی و هم ترازوی چندگانه دنباله ها و کاربرد های آن در انفورماتیک زیستی (۴ جلسه)
۴. درخت های تبار (Phylogenetic Trees) و هاپلو تایپ (Haplotyping) (۴ جلسه)
۵. توالی یابی DNA و پروتئین (۴ جلسه)
۶. پیش بینی ساختار مولکول های زیستی (۴ جلسه)
۷. تحلیل داده های ژنی و شبکه های زیستی (۴ جلسه)
۸. پیدا کردن ژن ها و موتیف ها (۴ جلسه)

مراجع اصلی

- Hans-Joachim Bockenhauer, D. Bongartz, *Algorithmic Aspects of Bioinformatics*, Springer, 2007.
- D. Gusfield, *Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology*, Cambridge University Press, 1997.
- N. C. Jones, P. A. Pevzner, *An Introduction to Bioinformatics Algorithms (Computational Molecular Biology)*, MIT Press, 2004.