



دانشگاه صنعتی شریف
دانشکده مهندسی کامپیوتر

برنامه درسی و ریز مواد دروس کارشناسی ارشد مهندسی کامپیوتر
گرایش نرم افزار - زیرگرایش مهندسی نرم افزار
* پیشنهاد اولیه *

نسخه ۴ - فروردین ۱۳۹۲

فهرست مطالب

اهداف، مبانی و ساختار کلی برنامه زیرگرایش مهندسی نرم افزار.....	۳
مشخصات و مقررات عمومی زیرگرایش مهندسی نرم افزار.....	۴
دروس اصلی (اجباری و اختیاری) زیر گرایش مهندسی نرم افزار.....	۴
دروس سمینار و پروژه زیرگرایش مهندسی نرم افزار.....	۵
دروس جبرانی زیرگرایش مهندسی نرم افزار.....	۵
نمونه هایی از برنامه های کارشناسی ارشد مهندسی نرم افزار.....	۶
ریز مواد دروس زیرگرایش مهندسی نرم افزار.....	۷
آداب فناوری اطلاعات (۴۰۳۴۷).....	۸
سیستمهای پشتیبانی تصمیم گیری (۴۰۵۴۸).....	۱۱
معماری نرم افزار (۴۰۶۴۶).....	۱۳
تولید برنامه از توصیف رسمی (۴۰۶۸۴).....	۱۵
متدولوژی های ایجاد نرم افزار (۴۰۷۲۴).....	۱۷
توصیف و واریسی برنامه ها (۴۰۷۴۵).....	۱۹
الگوها در مهندسی نرم افزار (۴۰۷۸۷).....	۲۱
آزمون نرم افزار (۴۰۸۲۸).....	۲۳
مهندسی نرم افزار ۲ (۴۰۹۲۴).....	۲۵
طراحی پایگاه داده ها ۲ (۴۰۹۳۸).....	۲۶
مهندسی نیازمندی ها (۴۰----).....	۲۸
تکامل نرم افزار (۴۰----).....	۳۰
ضمائم.....	۳۲
ضمیمه الف: ساختار کلی برنامه پیشنهادی ACM/IEEE برای کارشناسی ارشد مهندسی نرم افزار.....	۳۳
ضمیمه ب: مباحث بدنه اصلی دانش مهندسی نرم افزار – CBOK.....	۳۴

اهداف، مبانی و ساختار کلی برنامه زیرگرایش مهندسی نرم‌افزار

اهداف: زیرگرایش مهندسی نرم‌افزار با هدف تربیت نیروی متخصص در زمینه ایجاد نرم‌افزار در سطح کارشناسی ارشد، ذیل رشته مهندسی کامپیوتر - گرایش نرم‌افزار، تعریف شده است. دانشجویان این زیرگرایش ضمن گذراندن دروس تخصصی حوزه مهندسی نرم‌افزار با موضوعات پژوهشی و کاربردی این حوزه آشنا خواهند شد و پس از فارغ‌التحصیلی قادر خواهند بود تا آموخته‌های خود را در صنعت یا در راستای پژوهش، از جمله برای ادامه تحصیل در مقطع دکتری، به کار گیرند.

مبانی: برنامه زیرگرایش مهندسی نرم‌افزار بر مبنای برنامه پیشنهادی ACM/IEEE برای کارشناسی ارشد مهندسی نرم‌افزار (نسخه ۲۰۰۹)، و با استفاده از دروس موجود مهندسی نرم‌افزار در مقطع کارشناسی ارشد، تدوین شده است*. طبق برنامه پیشنهادی ACM/IEEE (که ساختار آن در ضمیمه الف آورده شده است)، حداکثر نیمی از واحدهای درسی باید به مباحث بدنه اصلی دانش مهندسی نرم‌افزار - CBOK[†] - تخصیص یابد؛ لذا آن دسته از دروس موجود که با حوزه‌های دانش CBOK (جدول الف) همپوشانی بیشتری دارند در مجموعه دروس اجباری قرار داده شده‌اند و دانشجویان ملزم شده که حداقل چهار درس خود را از بین دروس این مجموعه انتخاب کند (حوزه‌های دانش CBOK به تفصیل در ضمیمه ب توضیح داده شده‌اند).

ساختار: برنامه زیرگرایش مهندسی نرم‌افزار در مقطع کارشناسی ارشد برای یک دوره دوساله (مکعب از چهار نیمسال) طراحی شده است. تعداد واحدها برای دوره پژوهش-محور ۲۹ واحد و برای دوره آموزش-محور ۳۲ واحد است. دسته‌بندی دروس به قرار زیر است (اطلاعات کاملتر برنامه در بخشهای بعدی آورده شده است):

۱. دروس اصلی

- اجباری: ۱۲ واحد (چهار درس سه-واحدی)
- اختیاری: ۹ واحد (سه درس) برای دوره پژوهش-محور؛ ۱۲ واحد (چهار درس) برای دوره آموزش-محور
- ۲. پروژه کارشناسی ارشد: ۶ واحد (مختص دانشجویان پژوهش-محور)؛ دانشجویان آموزش-محور باید به جای ۶ واحد پروژه، دو درس سه-واحدی اخذ کنند.
- ۳. سمینار کارشناسی ارشد: ۲ واحد

جدول الف. گروه‌بندی حوزه‌های دانش در بدنه اصلی دانش مهندسی نرم‌افزار (CBOK)

مشخصه	عنوان حوزه دانش (فارسی)	عنوان حوزه دانش (لاتین)
A	آداب و رفتار حرفه‌ای	Ethics and Professional Conduct
B	مهندسی سیستم	System Engineering
C	مهندسی خواسته‌ها	Requirements Engineering
D	طراحی نرم‌افزار	Software Design
E	ساخت نرم‌افزار	Software Construction
F	آزمون	Testing
G	مراقبت و نگهداری از نرم‌افزار	Software Maintenance
H	مدیریت پیکربندی	Configuration Management
I	مدیریت مهندسی نرم‌افزار	Software Engineering Management
J	فرایند مهندسی نرم‌افزار	Software Engineering Process
K	کیفیت نرم‌افزار	Software Quality

* این برنامه در سایت اینترنتی روبرو موجود است: <http://www.gswe2009.org/curriculum/recommendations>

[†] Core Body of Knowledge

مشخصات و مقررات عمومی زیرگرایش مهندسی نرم‌افزار

- **طول دوره:** طول دوره دو سال است.
- **واحدها:** تعداد واحدهای دوره پژوهش-محور ۲۹ واحد و دوره آموزش-محور ۳۲ واحد است (دانشجویان حق ندارند دروسی را که قبلاً در دوره کارشناسی گرفته‌اند مجدداً اخذ نمایند). دسته‌بندی دروس به قرار زیر است:
 ۱. **دروس اصلی**
 - **اجباری:** ۱۲ واحد - از جدول ۱
 - **اختیاری:** ۹ واحد (برای پژوهش-محور) یا ۱۲ واحد (برای آموزش-محور) - از جداول ۱ و ۲
 ۲. **پروژه کارشناسی ارشد:** ۶ واحد - مطابق جدول ۳ (مختص دانشجویان پژوهش-محور)؛ دانشجویان آموزش-محور باید به جای شش واحد پروژه، دو درس سه-واحدی اخذ کنند. مهلت تصویب تعریف پروژه کارشناسی ارشد، پایان نیمسال دوم می باشد.
 ۳. **سمینار کارشناسی ارشد:** ۲ واحد - مطابق جدول ۳؛ دانشجو باید در ترم سوم و پس از تصویب پروژه، سمیناری در زمینه پروژه ارائه دهد که گزارش پیشرفت پروژه محسوب می‌شود. سمینار دانشجویان آموزش-محور نیز باید در ترم سوم و پس از تأیید موضوع توسط گروه، اخذ شود.
- **دروس جبرانی:** ممکن است دانشجو غیر از دروس اصلی، ملزم به گذراندن دروس جبرانی نیز باشد. در صورتی که گذراندن دروس جبرانی به تشخیص مدیر گروه برای دانشجو ضروری باشد، می‌بایستی حداکثر تا پایان سال اول تحصیل با تأیید گروه آموزشی حداکثر چهار درس از دروس جدول ۴ را بگذراند (این دروس، همان دروس جبرانی گرایش نرم‌افزار هستند). اخذ ۸ واحد جبرانی یا بیشتر امکان افزایش حداکثر یک نیمسال تحصیلی را به سنوات تحصیلی دانشجو فراهم می کند. حداقل نمره قبولی در درس جبرانی ۱۲ است.
- **همکاری با دانشکده:** کلیه دانشجویان نوبت اول باید به صورت TA یا با انجام امور ارجاعی دیگر (معادل ۲ واحد) با دانشکده همکاری کنند. همکاری با دانشکده برای دانشجویان آموزش-محور نیز اجباری است.
- **استاد راهنما:** مدیر گروه تا قبل از انتخاب و تصویب پروژه استاد راهنمای دانشجویان است. پس از تصویب تعریف پروژه، استاد راهنمای پروژه هر دانشجو مسئول تأیید و راهنمایی وی در اخذ واحدهای درسی می باشد.

دروس اصلی (اجباری و اختیاری) زیرگرایش مهندسی نرم‌افزار

جدول ۱. لیست دروس اصلی اجباری زیرگرایش - حداقل چهار درس باید گذرانده شود

ردیف	شماره درس	تعداد واحد	نام درس (فارسی)	نام درس (لاتین)	حوزه دانش متناظر* (CBOK)
۱	۴۰۳۴۷	۳	آداب فناوری اطلاعات	Information Technology Ethics	A (Ethics)
۲	۴۰۵۴۸	۳	سیستم‌های پشتیبانی تصمیم‌گیری	Decision Support Systems	B (Sys. Eng.)
۳	۴۰۶۴۶	۳	معماری نرم‌افزار	Software Architecture	D (Design)
۴	۴۰۷۲۴	۳	متدولوژی‌های ایجاد نرم‌افزار	Software Development Methodologies	I (Process)
۵	۴۰۷۴۵	۳	توصیف و واری برنام‌ها	Program Specification and Verification	K (Quality)
۶	۴۰۸۲۸	۳	آزمون نرم‌افزار	Software Testing	F (Testing)
۷	۴۰۹۲۴	۳	مهندسی نرم‌افزار ۲	Software Engineering II	I (Management)
۸	-----	۳	مهندسی خواسته‌ها	Requirements Engineering	C (Req. Eng.)
۹	-----	۳	تکامل نرم‌افزار	Software Evolution	G (Maintenance)

* مطابق با حوزه‌های دانش تعریف شده در برنامه پیشنهادی ACM/IEEE برای کارشناسی ارشد مهندسی نرم‌افزار (نسخه ۲۰۰۹)

جدول ۲. لیست دروس اصلی اختیاری زیرگرایش

ردیف	شماره درس	تعداد واحد	نام درس (فارسی)	نام درس (لاتین)
۱	۴۰۶۸۴	۳	تولید برنامه از توصیف رسمی	Formal Program Development
۲	۴۰۷۳۷	۳	مفاهیم پیشرفته معماری نرم افزار	Advanced Topics in Software Architecture
۳	۴۰۷۸۷	۳	الگوها در مهندسی نرم افزار	Patterns in Software Engineering
۴	۴۰۹۲۵	۳	مفاهیم پیشرفته مهندسی نرم افزار	Advanced Topics in Software Engineering
۵	۴۰۹۲۶	۳	مفاهیم پیشرفته سیستم های نرم افزاری	Advanced Topics in Software Systems
۶	۴۰۹۳۸	۳	طراحی پایگاه داده ها ۲	Database Design II
۷	حداکثر یک درس از دروس زیرگرایش های دیگر نرم افزار با موافقت استاد راهنما (دانشجویان آموزش محور می توانند حداکثر سه درس اخذ نمایند)			
۸	حداکثر یک درس از رشته های دیگر دانشکده یا دانشگاه مرتبط با رساله با موافقت استاد راهنما (پس از اخذ پروژه)			

دروس سمینار و پروژه زیرگرایش مهندسی نرم افزار

جدول ۳. لیست دروس پروژه و سمینار

ردیف	شماره درس	تعداد واحد	نام درس (فارسی)	نام درس (لاتین)
۱	۴۰۶۶۰	۶	پایان نامه کارشناسی ارشد	M.Sc. Thesis
۲	۴۰۹۰۰ (گروه ۱)	۲	سمینار کارشناسی ارشد	M.Sc. Seminar

دروس جبرانی زیرگرایش مهندسی نرم افزار

جدول ۴. لیست دروس جبرانی زیرگرایش مهندسی نرم افزار

ردیف	شماره درس	تعداد واحد	نام درس (فارسی)	نام درس (لاتین)
۱	۴۰۲۵۴	۳	ساختمان داده ها و الگوریتم ها	Data Structures and Algorithms
۲	۴۰۳۵۴	۳	طراحی الگوریتم ها	Design of Algorithms
۳	۴۰۴۱۴	۳	طراحی کامپایلرها	Compiler Design
۴	۴۰۴۲۴	۳	سیستم های عامل	Operating Systems
۵	۴۰۴۷۴	۳	مهندسی نرم افزار	Software Engineering

نمونه‌هایی از برنامه‌های کارشناسی ارشد مهندسی نرم‌افزار

دوره کارشناسی ارشد مهندسی نرم‌افزار در دانشگاه‌های متعدد دنیا به صورت رشته مستقل یا گرایش ارائه می‌شود. نمونه‌هایی از این برنامه‌ها در ادامه ذکر شده‌اند.

1. Carnegie-Mellon University, School of Computer Science (USA)
Degree Program: Master of Software Engineering
Webpage: <http://mse.isri.cmu.edu/software-engineering/web3-programs/MSE/index.html>
Curriculum: <http://mse.isri.cmu.edu/software-engineering/web3-programs/MSE/Plan of Study.html>
2. University of Texas at Austin, Department of Electrical and Computer Engineering (USA)
Degree Program: Master of Science in Electrical and Computer Engineering (Software Engineering Track)
Webpage: http://www.ece.utexas.edu/graduate/academic_tracks.cfm?id=11
Curriculum: <http://www.edge.utexas.edu/se/policies.html>
3. University of Southern California, Department of Computer Science (USA)
Degree Program: Master of Science in Computer Science with Specialization in Software Engineering
Webpage and Curriculum: <http://www.cs.usc.edu/academics/masters/ms-software-engineering.htm>
4. University of California at Irvine, School of Information and Computer Sciences (USA)
Degree Program: Master of Science in Software Engineering
Webpage and Curriculum: http://www.ics.uci.edu/grad/degrees/degree_se.php
5. Imperial College London, Department of Computing (UK)
Degree Program: Master of Science in Computing with Specialization in Software Engineering
Webpage: <http://www3.imperial.ac.uk/computing/admissions/pg/specialism>
Curriculum: <http://www3.imperial.ac.uk/computing/teaching/pg/mcsse>
6. University of Oxford, Department of Computer Science (UK)
Degree Program: Master of Science in Software Engineering
Webpage: <http://www.cs.ox.ac.uk/softeng/>
Curriculum: <http://www.cs.ox.ac.uk/softeng/courses/subjects.html>
7. Swiss Federal Institute of Technology (ETH Zürich), Department of Computer Science (Switzerland)
Degree Program: Master of Science in Computer Science (Software Engineering Track)
Webpage and Curriculum: http://www.inf.ethz.ch/education/master/master_CS/SE
8. University of Amsterdam, Department of Computer Science (Netherlands)
Degree Program: Master of Science in Software Engineering
Webpage: <http://www.uva.nl/en/disciplines/computer-science/education>
Curriculum: <http://gsi.uva.nl/programmes/content7/study-programme/study-programme.html>
9. Chalmers University of Technology, Department of Computer Science and Engineering (Sweden)
Degree Program: Master of Science in Software Engineering
Webpage: <http://www.chalmers.se/cse/EN/education/masters-programmes>
Curriculum: <http://www.chalmers.se/en/education/programmes/masters-info/Pages/Software-Engineering.aspx>
10. University of Melbourne, School of Engineering (Australia)
Degree Program: Master of Engineering (Software)
Webpage: <http://coursesearch.unimelb.edu.au/grad/1580-master-of-engineering>
Curriculum: <http://www.eng.unimelb.edu.au/study/graduate/master-eng-software.html>

ریز مواد دروس زیرگرایش مهندسی نرم افزار



گرایش: نرم افزار	مقطع: کارشناسی / تحصیلات تکمیلی
واحد: ۳	نوع درس: اجباری (زیرگرایش مهندسی نرم افزار)
پیش نیاز: -	هم نیاز: -

کلیات

فناوری اطلاعات به عنوان یک فناوری نو و پرکاربرد، واجد نقاط قوت و ضعف در بکارگیری و فرصت و تهدید برای کاربران است، و در عین حال به عنوان یک فناوری پر مصرف در حوزه حیات فردی و اجتماعی، غیرخنثی و تاثیرگذار است. جنبه‌های حرفه‌ای فعالیت‌های آن با گسترش بکارگیری اهمیت ویژه‌ای یافته است. بنابراین نظیر سایر فناوری‌ها در تولید، ساخت، بکارگیری مصرف چنانچه از آداب فنی متأثر از اخلاق فناورانه بهره‌نگیرد، از وجوه گوناگون از جمله توان جذب در زندگی فردی و اجتماعی شهروندان با دشواری مواجه می‌شود و عملاً کارائی خود را از دست می‌دهد. بنابراین مهندسان رایانه و فناوری اطلاعات نظیر سایر مهندسان نیازمند کسب سواد آداب حرفه‌ای و اخلاق فناوری اطلاعات به شکل منضبط و در قالب درس‌های دانشگاهی هستند.

آشنائی دانشجویان رایانه و فناوری اطلاعات با تحلیل‌های راهبردی این فناوری جهت شناسائی نقاط قوت و ضعف و فرصت و تهدیدات ناشی از بکارگیری گسترده آن، آشنائی با الگوهای مهندسی، حرفه‌ای و صنفی انجام کار، ضوابط حقوقی و قانونی فعالیت‌ها، تبعات اخلاقی و حرفه‌ای اعمال و مسئولیت‌ها، جرائم و خسارات و مجازات‌های رایانه‌ای، حقوق اطلاعاتی و گردش آزاد اطلاعات، حقوق تولید، تکثیر و ابداع محصولات فناوری اطلاعات، مفاهیم حریم شخصی و امنیت ملی و اطلاعات محرمانه و غیرمحرمانه در دسترسی و اشاعه آزاد اطلاعات و توانائی برآورد تبعات بکارگیری این فناوری از منظر موازین اخلاقی و معیارهای آدابی و فعالیت در یک محیط رایانشی امن از اهداف این درس است.

ریز مواد

۱. مقدمه

- هستان‌شناسی و تحلیل سوات (فرصت، تهدید، قوت و ضعف: SWAT) فناوری اطلاعات برای شخصیت‌های حقیقی و حقوقی مولد، کاربر و متأثر از آن

- چارچوب زکمن و استفاده از آن برای تعیین دانش اخلاقی مورد نیاز گروه‌های ذینفع

۲. نسبت آداب و اخلاق و جایگاه آنها در قبال قانون

- مروری بر تناسبات، تفاوت‌ها و همسانی‌های آداب با اخلاق
- مؤید و مراعی بودن و تعارض ناپذیری آداب در قبال قانون
- پیشینه‌گرایی قانون و آینده‌تایی آداب

۳. پیشینه و انواع نظامات ارزشی اخلاقی حامی آداب و نقاط قوت و ضعف آنها

- پیشینه فلسفی اخلاق و انواع نظام‌های اخلاقی مطلق و نسبی‌گرا، انسان‌گرا، فایده‌گرا، مصلحت‌گرا، وظیفه‌گرا، شک‌گرا، عینی‌گرا، ذهنی‌گرا، طبیعت‌گرا، عقل‌گرا، عدالت‌گرا و فضیلت‌گرا
- اخلاق در غرب
- اخلاق اسلامی

- اخلاق ایرانی و ایرانیان
- اخلاق جهانی و قوانین طلایی و عرفی اخلاقی
- ۴. تعارضات، بلا تکلیفی و تنگناهای اخلاق و راه حل های مواجهه با این وضعیت ها
 - تنگنای اخلاقی
 - تعارض اخلاقی
 - بلا تکلیفی اخلاقی
 - الگوهای حل بلا تکلیفی و داور اخلاقی
 - منطق موقعیت و تحلیل آن
 - از اخلاق گرایی تا اعتماد آفرینی
- ۵. ماهیت توافقی آداب، ساختار و کاربردها و کارکردهای انواع اسناد آدابی شامل آداب نامه ها، اساسنامه ها و منشورها
 - تدوین توافقی اسناد آدابی
 - تعریف، کاربرد و نحوه تدوین اساسنامه ها
 - تعریف، کاربرد، مدل و نحوه تدوین منشورهای آدابی
 - تعریف، کاربرد، مدل و نحوه تدوین آداب نامه ها
- ۶. مدل های مولد و حامی انواع الگوهای آداب نامه نویسی بویژه آداب نامه های سنجش پذیر
 - اسناد آدابی گروهی
 - اسناد آدابی بنگاهی
 - اسناد آدابی بنگاهی با امکان توفیق سنجی
 - بهبود مدل آداب نامه ها
- ۷. آداب نامه های فناوری اطلاعات و کاربردهای آن در حوزه های حرفه ای گری مهندسی حرفه ای فناوری اطلاعات و جایگاه نظام مهندسی فناوری اطلاعات در حوزه های حرفه ای گری
 - آداب نامه های حرفه ای
 - مهندسی حرفه ای فناوری اطلاعات
 - نظامات حرفه ای و مهندسی
 - نظام مهندسی فناوری اطلاعات، اهداف، کارکردها و ضرورت ها
 - نظامات صنفی، حرفه ای و مهندسی فناوری اطلاعات در جهان و ایران
- ۸. حق فکری، حق مالی، حق تکثیر و حق اختراع در حوزه فناوری اطلاعات، قوانین حامی، آداب نامه های مکمل و آینده آنها
 - حق مالی و فکری ابداعات
 - حق اختراع و تکثیر ابداعات
 - قوانین حامی در ایران و جهان
 - آداب نامه های حامی حقوق ابداعات
 - آینده این حقوق در جهان مجازی
- ۹. ابعاد اخلاقی جامعه اطلاعاتی و مجازی
 - گردش آزاد اطلاعات و حقوق شهروندی
 - امنیت، شفاف سازی و محرمانگی اطلاعات، آلودگی و تضمین کیفیت اطلاعات
 - آداب خلوت و حریم شخصی در رایاسپهر از جمله شبکه های اجتماعی
 - حریم شخصی در رایاسپهر و آینده آن
 - متافیزیک واقعیت مجازی و نقد اثرات فناوری اطلاعات

- جامعه عریان و مقابله با ترس از فناوری
- زندگی دوم، جهان مجازی، تغییر الگوی ارتباطی و تبعات آن بر نظامات کلان از جمله نظامات آموزشی
- سایت‌ها، وب نوشته‌ها، شبکه‌های اجتماعی، نیازها و تبعات، دشواری‌ها و راه حل‌ها
- مردم‌سالاری الکترونیکی و ابعاد آن
- سواد بهداشت و آداب فناوری اطلاعات برای همه
- ۱۰. انواع تخلفات و جرائم رایانه‌ای، پلیس اینترنتی، امنیت اجتماعی و شهروندی در رایاسپهر و راه های مقابله
- بدافزارها، ویروسها، کرمها، روش‌ها، حفره ها و اشکال نفوذ
- هکرها، کراکرها، آزاردهنده ها، مجرمین و تروریست های سایبری
- ارزشهای پیشگیرانه آداب فناوری اطلاعات در کاهش جرائم و تخلفات رایانه‌ای
- حمایت از گروه‌های سنی آسیب‌پذیر در مواجهه با محتوای رقمی آسیب‌رسان
- قوانین جرائم رایانه‌ای در ایران و جهان و حوزه‌های فردی، اجتماعی و ملی آسیب
- دلایل رقمی، کارشناسی رسمی فناوری اطلاعات و لزوم مشی‌های امنیتی سازمانی
- امنیت در بنگاه‌ها، سازمان‌های عمومی، خدمات و سامانه‌های الکترونیکی و نحوه تامین آن
- نرم افزارهای متن‌باز، انواع مجوزهای آن و حل دشواری‌های مالکیت ارزش‌افزای محصولات متکامل و سهم شالوده‌سازان
- حفظ امنیت با پایش گسترده یا حریم شخصی ناامن
- انواع قوانین فناوری اطلاعات در ایران و نقد آنها
- کیفیت زندگی و نقش فناوری اطلاعات و آداب فناوری اطلاعات سبز در زندگی اول و دوم رایانه‌ای و بازیافت ابزار مستهلک رایانه‌ای

آزمون - تمرین

- تمرین‌ها: ۳۵٪ کل نمره
- کوئیز: ۱۰٪ کل نمره
- آزمون میانی: ۲۵٪ کل نمره
- آزمون پایانی: ۳۰٪ کل نمره

مراجع اصلی

- G. Reynolds, *Ethics in Information Technology*, 4th edition, Thomson, 2012.
- I. Van De Poel, *Ethics, Technology, and Engineering*, Wiley-Blackwell, 2011.
- J. Harris, M.J. Rabins, C.E. Harris, *Engineering Ethics: Concepts and Cases*, Thomson, 2004.
- M.W. Martin, *Ethics in Engineering*, McGraw-Hill, 2005.
- D. Langford, *Internet Ethics*, Macmillan Press, 2000.

مراجع فرعی

- جمال خانی جزنی، *اخلاق و فناوری اطلاعات*، مرکز تحقیقات مخابرات ایران، ۱۳۸۵.
- جمال خانی جزنی، *اخلاق در فضای مجازی*، سازمان پژوهش‌های علمی و صنعتی ایران، ۱۳۸۷.
- سید ابراهیم ابطحی، چارچوب بهنجار آموزش اخلاق رایاسپهری: شالوده‌ای پویا برای طرح درس دانشگاهی آموزش آداب فناوری اطلاعات، دومین همایش منطقه ای اخلاق و فناوری اطلاعات، مرکز تحقیقات مخابرات ایران، ۱۳۸۵.
- احد فرامرز قراملکی، *اخلاق حرفه‌ای*، ناشر مولف، چاپ سوم، ۱۳۸۵.
- کامران شیرزاد، *جرائم رایانه‌ای از دیدگاه حقوق جزای ایران و حقوق بین‌الملل*، شرکت نشر بهینه فراگیر، ۱۳۸۸.



سیستم‌های پشتیبانی تصمیم‌گیری (۴۰۵۴۸)

Decision Support Systems

طراح درس: جعفر حبیبی

مقطع: تحصیلات تکمیلی	گرایش: نرم‌افزار
نوع درس: اجباری (زیرگرایش مهندسی نرم‌افزار)	واحد: ۳
هم‌نیاز: -	پیش‌نیاز: -

کلیات

سیستم‌های پشتیبانی تصمیم‌گیری واحد درسی است که طی آن فرایندهای تصمیم‌گیری و استفاده از ابزارهای کامپیوتری برای حل بهتر مسائل و غلبه بر پیچیدگی‌های موجود در آن‌ها معرفی می‌شود. در این درس دانشجویان با سیستم‌های پشتیبانی مدیریت (MSS)، پشتیبانی تصمیم‌گیری (DSS) و مکانیزم‌های مرتبط با هوشمندی تجاری (BI) و توسعه و ساخت این سیستم‌ها آشنا می‌شوند. در این درس چهارچوبی تدوین شده که دانشجویان در آن با ساختار سیستم‌های اطلاعاتی و ابزارها و تکنیک‌هایی نظیر انبار داده‌ها در فرایندهای سیستم پشتیبانی تصمیم‌گیری آشنا شده و در طی آن مهارت‌های لازم برای توسعه یک سیستم پشتیبانی تصمیم‌گیری را فرا می‌گیرند.

ریز مواد

۱. مفاهیم پایه فرایند تصمیم‌گیری
۲. نیاز به سیستم‌های پشتیبانی تصمیم‌گیری
۳. مفاهیم مطرح در هوشمندی تجاری (Business Intelligence) و رابطه آن با سیستم‌های پشتیبانی تصمیم‌گیری
۴. مدیران و تصمیم‌گیری، دورنمایی از سیستم‌های پشتیبانی مدیریت (MSS) شامل سیستم پشتیبانی تصمیم‌گیری (DSS)
۵. سیستم پشتیبانی تصمیم‌گیری گروهی (G-DSS)
۶. سیستم پشتیبانی اجرایی (EIS)
۷. سیستم خبره (ES)
۸. سیستم مدیریت دانش (KMS)
۹. مفهوم سیستم، تصمیم‌گیری، و فازهای آن
۱۰. پشتیبانی، دورنمایی از سیستم‌های پشتیبانی تصمیم‌گیری (DSS) شامل زیر سیستم داده، زیرسیستم مدل و زیرسیستم واسط کاربر
۱۱. مدیریت داده‌ها شامل مخزن داده‌ها، بازیابی، پایگاه داده‌ها در DSS، پردازش تحلیلی On Line (OLAP)، داده کاوی و نمایش
۱۲. انبار داده‌ها و مفاهیم مرتبط با آن
۱۳. مدل‌سازی و تحلیل شامل انواع مدل‌های ایستا و پویا، مدل‌سازی به وسیله صفحه گسترده، درخت تصمیم و جدول تصمیم، مدل‌های بهینه، مدل‌های ابتکاری (Heuristic)، شبیه‌سازی، مدل‌های چندبعدی (OLAP)، مدل‌سازی و شبیه‌سازی بصری
۱۴. توسعه DSS شامل چرخه سنتی عمر نرم‌افزار، و متدولوژی‌های مختلف برای توسعه DSS
۱۵. سیستم‌های پشتیبانی تصمیم‌گیری گروهی و تکنولوژی‌های پشتیبانی گروهی
۱۶. سیستم‌های پشتیبانی اجرایی EIS شامل سیستم‌های اطلاعاتی اجرایی، نقش مدیران و اطلاعات لازم، مشخصه‌های EIS، مقایسه EIS و DSS، زنجیره تأمین و پشتیبانی تصمیم‌گیری

۱۷. سیستم‌های خبره و هوش مصنوعی شامل مفهوم هوش مصنوعی و اجزای آن، مفهوم سیستم‌های خبره، ساختار سیستم‌های خبره، زمینه‌های سیستم‌های خبره، و انواع سیستم‌های خبره
۱۸. پشتیبانی تصمیم‌گیری شبکه‌ای مثل اینترنت، اینترنت و اکسترانت
۱۹. پیاده سازی DSS و استراتژی‌های آن
۲۰. DSS‌های هوشمند و آثار اجتماعی و سازمانی سیستم‌های پشتیبانی مدیریت (MSS)
۲۱. انبارهای داده و داده کاوی
۲۲. مدیریت ارتباط با مشتری (CRM)
۲۳. برنامه‌ریزی منابع سازمان (ERP)
۲۴. برنامه‌ریزی سلسه مراتبی (AHP)

آزمون - تمرین - گزارش پژوهشی

• تمرین‌های پژوهشی:

- (۱) تمرین‌هایی که باید با استفاده از اینترنت و متن کتاب تجزیه و تحلیل شده و پاسخ داده شوند.
- (۲) تمرین‌هایی که در ارتباط با استفاده از مفاهیم و ابزارهای موجود در دنیای واقعی است که تحت عنوان مطالعه موردی به دانشجویان داده می‌شود.
- **تمرین‌های عملی:** در این بخش دانشجویان موظف به اجرا و پیاده سازی پروژه‌های مرتبط با این درس و استفاده از ابزارهای متنوع متن - باز موجود در این زمینه هستند.
- **سمینار و ارائه کارهای پژوهشی:** در این درس دانشجویان به صورت گروهی و یا انفرادی باید کارهای پژوهشی خود را ارائه دهند و از سمینارهای مذکور ارزشیابی به عمل می‌آید.
- **آزمون‌های میانی و پایانی:** این درس شامل یک آزمون میانی است که در اواسط ترم برگزار شده و شامل مطالب ۶ فصل ابتدایی است. آزمون پایانی شامل کلیه مطالب تدریس شده است.

مراجع اصلی

- E. Turban, R. Sharda, D. Delen, *Decision Support System and Business Intelligent Systems*, 9th edition, Prentice Hall, 2010.
- G.M. Marakas, *Decision Support in the 21st Century*, 2nd edition, Prentice Hall, 2003.
- C. Todman, *Designing a Data Warehouse: Supporting Customer Relationship Management*, Prentice Hall Professional Technical Reference, 2000.
- S.A. Brown, *Customer Relationship Management: A Strategic Imperative in the World of e-Business*, John Wiley and Sons, 1999.



مقطع: تحصیلات تکمیلی	گرایش: نرم افزار
نوع درس: اجباری (زیرگرایش مهندسی نرم افزار)	واحد: ۳
هم نیاز: -	پیش نیاز: -

کلیات

هدف اصلی از این درس آشنایی با مفاهیم معماری نرم افزار، جایگاه آن، فرایند تدوین معماری، مستندسازی و ارزیابی معماری است که شامل اهداف جزئی زیر می باشد:

- فهم تاثیر پیشران های معماری بر ساختار سیستم های نرم افزاری
- درک نقش فنی، سازمانی و تجاری معماری نرم افزار
- شناخت ساختارهای کلیدی معماری (سبک ها، تاکتیک ها و ...)
- فهم اصول صحیح مستندسازی و ارائه معماری
- درک تاثیر COTS در طراحی معماری
- آشنایی با مشخصه های کیفی و روش های ارزیابی معماری
- آگاهی از آینده معماری نرم افزار

ریز مواد

۱. تعاریف معماری نرم افزار، پیشران های معماری
۲. چرخه حیات معماری
۳. نیازمندی های وظیفه مندی و غیر وظیفه مندی
۴. نقش معماری نرم افزار در دستیابی به ویژگی های کیفی نرم افزار
۵. جایگاه معماری نرم افزار در فرایند توسعه محصولات نرم افزاری
۶. ساختارها و منظرهای معماری
۷. تکنیک ها و متدهای طراحی معماری
۸. تاکتیک ها، الگوها و سبک های معماری
۹. طراحی معماری و روش های دستیابی به خصوصیات کیفی
۱۰. مستندسازی معماری و زبان های توصیف معماری
۱۱. روش های ارزیابی معماری نرم افزار
۱۲. ارزیابی معماری نرم افزار
۱۳. معماری نرم افزار خاص دامنه (DSSA)

- ۱۴. معماری خط تولید نرم افزار
- ۱۵. معماری سرویس گرا
- ۱۶. توسعه بر پایه معماری مولفه محور
- ۱۷. محاسبات ابری
- ۱۸. مباحث نوین در معماری نرم افزار
- ۱۹. آینده معماری نرم افزار

آزمون - تمرین - پروژه - سمینار

- تمرینات: ۱۵٪ کل نمره
- پروژه: ۲۵٪ کل نمره
- آزمون پایان ترم: ۵۰٪ کل نمره
- سمینار: ۱۰٪ کل نمره

مراجع اصلی

- M. Shaw, D. Garlan, P. Hall, *Software Architecture: Perspectives on an Emerging Discipline*, Prentice-Hall, 1996.
- L. Bass, P. Clements, R. Kazman, *Software Architecture in Practice*, 2nd edition, Addison-Wesley, 2003.
- P. Clements, et al. *Documenting Software Architectures: Views and Beyond*, Addison-Wesley, 2003.
- A.J. Lattanze, *Architecting Software Intensive Systems: A Practitioner's Guide*, Auerbach Publications, 2008.



تولید برنامه از توصیف رسمی (۴۰۶۸۴)

Formal Program Development

طراح درس: سید حسن میریان حسین آبادی

مقطع: تحصیلات تکمیلی	گرایش: نرم افزار
نوع درس: اختیاری (زیرگرایش مهندسی نرم افزار)	واحد: ۳
هم نیاز: -	پیش نیاز: -

کلیات

این درس برای دانش جویان کارشناسی ارشد و دکتری ارایه می شود و هدف از آن پرداختن به روش های تولید برنامه از توصیف رسمی سیستم ها به طور سیستماتیک است. در این درس تولید برنامه از توصیف نوشته شده به زبان Z و به خصوص تقلید و جبر پالایش مورد بحث قرار می گیرد. روش های دیگر از جمله جبر پالایش مورگان، متدولوژی B، تئوری انواع (Type Theory) و نسخه ساختنی Z معرفی می گردند. ضمناً ابزارهای لازم برای به کارگیری این روش ها معرفی می گردد.

ریز مواد

۱. مقدمه ای بر تولید برنامه
 - a. چرا تولید برنامه از توصیف رسمی؟
 - b. تولید برنامه (تبدیل، پالایش، تقلید و تئوری انواع)
۲. تولید برنامه از توصیف نوشته شده به زبان Z (۳ هفته)
 - a. تقلید (Animation) با استفاده از زبان های تابعی و منطقی
 - b. معرفی ابزارهای تقلید
 - c. پالایش (Refinement)
 - d. معرفی ابزارهای پالایش
۳. جبر پالایش مورگان (۴ هفته)
 - a. برنامه ها و پالایش
 - b. انواع و اعلان ها
 - c. جایگزینی و ترکیب ترتیبی
 - d. جملات انتخابی
 - e. ثابت های منطقی
 - f. حلقه های تکرار
 - g. رویه ها و پارامترها
 - h. مثال
۴. متدولوژی B (۳ هفته)
 - a. ماشین های انتزاعی
 - b. ساختن توصیف

c. طراحی و پالایش

d. اثبات و پیاده‌سازی

e. مثال

۵. تئوری انواع (۳ هفته)

a. مقدمه‌ای بر ریاضیات ساختنی (Constructive Mathematics)

b. تئوری انواع Martin-Löf

c. تولید برنامه با استفاده از تئوری انواع

d. مثال

۶. نسخه ساختنی زبان توصیف Z (CZ) (۲ هفته)

a. انواع تئوری مجموعه‌های ساختنی

b. اصول CZ

c. جبر شمای ساختنی

d. تولید برنامه در CZ

e. مثال

آزمون - تمرین - پروژه - سمینار

- دانشجویان به گروه‌های ۲ یا ۳ نفره تقسیم می‌شوند و هر گروه سه صورت برنامه در اندازه‌های کوچک، متوسط و بزرگ را پیشنهاد می‌نماید. پس از تصویب برنامه‌ها، هر گروه تمرینات (حداقل ۳ تمرین) را در طول ترم براساس مسایل پیشنهادی خود پاسخ خواهد داد.
- برنامه بزرگ صورت پروژه هر گروه را مشخص می‌کند که یک ماه پس از پایان امتحانات فرصت دارند تا پروژه خود را تحویل نمایند. دانشجویان باید حتی‌الامکان با استفاده از نرم‌افزارهای موجود در این زمینه کار نمایند.
- دانشجویان به طور اختیاری سمیناری را در ارتباط با مطالب درس پس از گرفتن تایید ارایه می‌نمایند.
- حدود ۷ آزمون کوچک به عنوان امتحان میان ترم
- آزمون نهایی

مراجع اصلی

- C. Morgan, *Programming from Specifications*, Prentice Hall, 1990.
- A. Diller, *Z: An Introduction to Formal Methods*, John Wiley and Sons, 1992.
- K. Lano, H. Haughton, *Specification in B: An Introduction Using the B Toolkit*, Imperial College Press, 1996.
- B. Nordstrom, K. Petersson, J.M. Smith, *Programming in Martin Löf's Type Theory: An Introduction*, Oxford University Press, 1990.
- J. Woodcock, J. Davies, *Using Z Specifications, Refinement, and Proof*, Prentice Hall Europe, 1996.
- D. Gries, F.B. Schneider, *A Logical Approach to Discrete Math*, Springer Verlag, 1993.



متدولوژی‌های ایجاد نرم‌افزار (۴۰۷۲۴)

Software Development Methodologies

طراح درس: رامان رامسین

گرایش: نرم‌افزار	مقطع: تحصیلات تکمیلی
واحد: ۳	نوع درس: اجباری (زیرگرایش مهندسی نرم‌افزار)
پیش‌نیاز: -	هم‌نیاز: -

کلیات

هدف این درس، آشنا کردن دانشجویان کارشناسی ارشد و دکترا با متدولوژی‌های ایجاد نرم‌افزار و مفاهیم و اصول مرتبط است. دانشجویان ضمن آشنایی با متدولوژی‌های مطرح، با روش‌های تحلیل و ارزیابی متدولوژی‌ها، الگوها/پادالگوها و متامدل‌های فرایند ایجاد نرم‌افزار، و روش‌های مهندسی متدولوژی آشنا می‌شوند. این درس از نظر ساختار و محتوا متناظر با درس متدها می باشد که از طرف انستیتو مهندسی نرم‌افزار (SEI) پیشنهاد شده و از دروس اصلی کارشناسی ارشد مهندسی نرم‌افزار (MSE) در دانشگاه Carnegie Mellon است.

با توجه به اینکه در حال حاضر مشی شیء‌گرا در بین متدولوژی‌ها مبنای غالب است، ساختار و محتوای فعلی درس عمدتاً بر متدولوژی‌های شیء‌گرا تمرکز دارد.

ریز مواد

۱. مقدمه - معرفی تاریخچه تکاملی متدولوژی‌های شیء‌گرا و معیارهای ارزیابی مربوطه (۲ جلسه - مدت هر جلسه، نود دقیقه است)
۲. معرفی تحلیلی متدولوژی Fusion - بررسی نمودهای بارز مشی شیء‌گرا (۲ جلسه)
۳. معرفی اجمالی متدولوژی‌های شاخص نسل‌های اول و دوم - متدولوژی‌های OMT, Booch, RDD, Coad-Yourdon, BON, OOSE, و Hodge-Mock (۴ جلسه)
۴. معرفی تحلیلی متدولوژی‌های نسل سوم - متدولوژی‌های EUP, RUP/USDP, UML-Components, Catalysis, OPM, و OPEN, و FOOM (۷ جلسه)
۵. معرفی تحلیلی متدولوژی‌های چابک - متدولوژی‌های XP, Scrum, DSDM, Crystal, AUP, ASD, و FDD (۷ جلسه)
۶. معماری و ایجاد نرم‌افزار به روش مبتنی بر مدل - MDA و MDD (۱ جلسه)
۷. الگوها و پاد الگوهای فرایند ایجاد نرم‌افزار (۲ جلسه)
۸. متامدل‌های فرایند ایجاد نرم‌افزار (۱ جلسه)
۹. روش‌های مهندسی متدولوژی - تحلیل و طراحی (۲ جلسه)
۱۰. معرفی ابزار مهندسی متدولوژی EPFC (۲ جلسه)

آزمون - تمرین - گزارش پژوهشی

- آزمون: آزمون‌های میان ترم و پایان ترم (۶۰٪ کل نمره)
- تمرین: سه تمرین پژوهشی، و یک تمرین عملی مهندسی متدولوژی با ابزار EPFC؛ تمرینات در طول نیمسال تحویل داده می‌شوند (۲۵٪ کل نمره).
- گزارش پژوهشی: موضوع پژوهش قبل از آزمون پایان نیمسال تعیین می‌شود. دانشجو کار پژوهش را با کمک استاد درس آغاز کرده و پس از انجام کار، نتیجه پژوهش را در قالب گزارش ارائه می‌دهد (۱۵٪ کل نمره).

مراجع اصلی

- S.W. Ambler, *Process Patterns: Building Large-Scale Systems Using Object Technology*, Cambridge University Press, 1998.
- S.W. Ambler, J. Nalbone, M.J. Vizdos, *The Enterprise Unified Process: Extending the Rational Unified Process*, Prentice-Hall, 2005.
- A. Cockburn, *Agile Software Development: The Cooperative Game*, 2nd edition, Addison-Wesley, 2006.
- OMG, *Model Driven Architecture (MDA) Guide*, Object Management Group (OMG), 2003.
- OMG, *Software and Systems Process Engineering Metamodel Specification (v2.0)*, Object Management Group (OMG), 2007.
- J. Ralyté, S. Brinkkemper, B. Henderson-Sellers (Eds.), *Situational Method Engineering: Fundamentals and Experiences*, Springer, 2007.
- R. Ramsin, R.F. Paige, Process-centered review of object-oriented software development methodologies, *ACM Computing Surveys* 40, 1 (February), Article 3, 89 pages, 2008.
- P. Shoval, *Functional and Object Oriented Analysis and Design: An Integrated Methodology*, Idea Group Publishing, 2007.



توصیف و واریسی برنامه‌ها (۴۰۷۴۵)

Program Specification and Verification

طراح درس: سید حسن میریان حسین آبادی

گرایش: نرم‌افزار	مقطع: تحصیلات تکمیلی
واحد: ۳	نوع درس: اجباری (زیرگرایش مهندسی نرم‌افزار)
پیش‌نیاز: -	هم‌نیاز: -

کلیات

این درس برای دانش‌جویان کارشناسی ارشد و دکتری ارایه می‌شود و هدف از آن پرداختن به روش‌های صوری برای توصیف و واریسی سیستم‌ها است. در این درس ابزارهای لازم برای به‌کارگیری این روش‌ها معرفی و در مورد رابطه بین توصیف صوری و پیاده‌سازی به طور اختصار بحث می‌گردد.

ریز مواد

۱. مقدمه‌ای بر توصیف سیستم‌ها (۱ هفته)
 - (a) چرا توصیف صوری؟
 - (b) توصیف صوری و مهندسی نرم‌افزار
 - (c) تولید برنامه از توصیف (پالایش)
۲. جبر گزاره‌ها، جبر مسندات (۱ هفته)
۳. تئوری مجموعه‌ها و زبان Z (۲ هفته)
 - (a) تساوی
 - (b) انواع، مجموعه‌ها و عملیات روی آنها
 - (c) تعاریف
 - (d) روابط و عملیات روی آنها
 - (e) توابع و عملیات روی آنها
 - (f) مثال
۴. واحدهای ساختاری توصیف (۲ هفته)
 - (a) شِما (Schema) و نحوه مدل کردن سیستم
 - (b) استفاده از شِما به عنوان اعلان، نوع و مسند
 - (c) شِمای ژنریک
 - (d) نحوه بیان اصول (Axiomatic Description)
 - (e) مثال
۵. جبر شِماها (Schema Calculus) (۲ هفته)
 - (a) تغییر متغیر (Renaming and Decoration)
 - (b) ترکیب شِماها با استفاده از عملگرهای $\wedge, \vee, \neg, \Rightarrow, \exists, \forall, ;, \backslash$ و Inclusion
 - (c) مثال

۶. ابزارگان ریاضی Z (۱ هفته)

(a) ردیف‌ها و Bagها و عملیات روی آنها

(b) نوع آزاد (Free Type)

(c) مثال

۷. توصیف با استفاده از ارتقا (Promotion) (۱ هفته)

۸. امکان‌پذیری توصیف و محاسبه پیش شرطها (Precondition) (۱ هفته)

۹. واریسی (Verification) (۳ هفته)

(a) اصول تئوری مجموعه‌ها

(b) قوانین استنتاج

(c) قضیه حالت اولیه سیستم

(d) ساده‌سازی پیش شرطها

(e) اثبات خصوصیات توصیف

(f) مثال

۱۰. تولید برنامه از توصیف صوری Z با استفاده از پالایش (Refinement) (۲ هفته)

(a) پالایش ساختارهای داده‌ای

(b) پالایش عملیات

(c) مثال

آزمون - تمرین - پروژه - سمینار

- دانشجویان به گروه‌های ۲ یا ۳ نفره تقسیم می‌شوند و هر گروه سه صورت برنامه در اندازه‌های کوچک، متوسط و بزرگ را پیشنهاد می‌نماید. پس از تصویب برنامه‌ها، هر گروه تمرینات (حداقل ۳ تمرین) را در طول ترم براساس مسایل پیشنهادی خود پاسخ خواهد داد.
- برنامه بزرگ صورت پروژه هر گروه را مشخص می‌کند که یک ماه پس از پایان امتحانات فرصت دارند تا توصیف صوری کامل پروژه را تحویل نمایند.
- دانشجویان باید با استفاده از نرم‌افزارهای کنترل کننده جامعیت و عدم تناقض و اثبات قضیه خصوصیات توصیف صوری خود را مورد ارزیابی قرار دهند.
- دانش‌جویان به طور اختیاری سمیناری را در ارتباط با مطالب درس پس از گرفتن تایید ارایه می‌نمایند.
- حدود ۷ آزمون کوچک به جای امتحان میان ترم
- آزمون نهایی

مراجع اصلی

- J. Woodcock, J. Davies, *Using Z Specifications, Refinement, and Proof*, Prentice Hall Europe, 1996.
- D. Gries, F.B. Schneider, *A Logical Approach to Discrete Math*, Springer Verlag, 1993.
- C. Morgan, *Programming from Specifications*, Prentice Hall, 1990.



مقطع: تحصیلات تکمیلی	گرایش: نرم افزار
نوع درس: اختیاری (زیرگرایش مهندسی نرم افزار)	واحد: ۳
هم نیاز: -	پیش نیاز: -

کلیات

هدف این درس، آشنا کردن دانشجویان کارشناسی ارشد و دکترا با الگوها و کاربرد آنها در مهندسی نرم افزار است. دانشجویان ضمن آشنایی با الگوهای رایج تحلیل، طراحی، معماری، مهندسی مجدد و مهندسی فرایند، با الگوهای اصلاح کد و پادالگوها نیز آشنا می شوند. به دلیل تعدد الگوها، سعی میشود ضمن تأکید بر معرفی تفصیلی الگوهای پرکاربرد، آشنایی کافی با ساختارها و اصول مبنایی و روشهای مدیریت پیچیدگی و تحلیل الگوها نیز حاصل شود.

ریز مواد

۱. مقدمه: مبانی و تاریخچه (۱ جلسه - مدت هر جلسه، نود دقیقه است)
۲. الگوهای پایه Coad (۱ جلسه)
۳. الگوهای Gamma et al. - GoF (۵ جلسه)
۴. اصول و قواعد شیءگرایی در قالب الگوها - الگوهای GRASP (۲ جلسه)
۵. الگوهای معماری Buschmann et al. - GoV (۴ جلسه)
۶. الگوهای طراحی GoV (۲ جلسه)
۷. الگوهای اصلاح و بازآرایی کد (۴ جلسه)
۸. الگوهای مهندسی مجدد (۴ جلسه)
۹. الگوهای فرایند ایجاد نرم افزار (۱ جلسه)
۱۰. پادالگوها (۳ جلسه)
۱۱. الگوهای تحلیل Fowler (۲ جلسه)
۱۲. روشهای طبقه بندی، مدیریت پیچیدگی و تحلیل الگوها (۱ جلسه)

آزمون - تمرین - گزارش پژوهشی

- آزمون: آزمونهای میان ترم و پایان ترم (۶۰٪ کل نمره)
- تمرین: چهار یا پنج تمرین پژوهشی که در طول نیمسال تحویل داده می شوند (۲۰٪ کل نمره).
- گزارش پژوهشی: موضوع مقاله قبل از آزمون پایان نیمسال تعیین می شود. دانشجویان کار پژوهش را با کمک استاد درس آغاز می کنند. نتایج باید نهایتاً در قالب یک گزارش پژوهشی تهیه و تحویل داده شوند (۲۰٪ کل نمره).

مراجع اصلی

- F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, M. Stal, *Pattern-Oriented Software Architecture: A System of Patterns*, Vol. 1, Wiley, 1996.
- F. Buschmann, K. Henney, D.C. Schmidt, *Pattern-Oriented Software Architecture: On Patterns and Pattern Languages*, Vol. 5, Wiley, 2007.
- M. Fowler, *Analysis Patterns: Reusable Object Models*, Addison Wesley, 1996.
- E. Gamma, R. Helm, R. Johnson, J. Vlissides, *Design Patterns: Elements of Reusable ObjectOriented Software*, Addison Wesley, 1995.
- J. Kerievsky, *Refactoring to Patterns*, Addison Wesley, 2004.
- D. Manolescu, M. Voelter, J. Noble, *Pattern Languages of Program Design*, Vol. 5, Addison Wesley, 2006.
- A. Shalloway, J. Trott, *Design Patterns Explained: A New Perspective on Object-Oriented Design*, 2nd edition, Addison Wesley, 2005.

آزمون نرم افزار (۴۰۸۲۸)



دانشکده مهندسی کامپیوتر

Software Testing

طراح درس: سید حسن میریان حسین آبادی

گرایش: نرم افزار	مقطع: تحصیلات تکمیلی
واحد: ۳	نوع درس: اجباری (زیرگرایش مهندسی نرم افزار)
پیش نیاز: -	هم نیاز: -

کلیات

این درس برای دانش جویان کارشناسی ارشد و دکتری ارایه می شود و هدف از آن پرداختن به روش های تولید داده آزمون از روی مدل های مختلف نرم افزار به طور سیستماتیک است. همچنین در طی این درس ابزارهای لازم برای به کارگیری این روش ها معرفی می گردد.

ریز مواد

۱. مقدمه ای بر روش های آزمون (۲ هفته)
 - a. فعالیت های مهندس آزمون
 - b. استراتژی های آزمون (آزمون واحد، تجمیع)
 - c. محدودیت ها و اصطلاحات جدید در آزمون نرم افزار
 - d. نقش معیارهای پوشش در آزمون نرم افزار
 - e. اصطلاحات قدیمی در آزمون نرم افزار
 - آزمون جعبه سیاه و جعبه سفید
 - آزمون بالا به پایین و پایین به بالا
 - آزمون ایستا و پویا
۲. معیارهای پوشش (۹ هفته)
 - a. معیارهای مبتنی بر گراف (۳ هفته)
 - b. معیارهای مبتنی بر منطق (۲ هفته)
 - c. معیارهای مبتنی بر افراز فضای ورودی (۲ هفته)
 - d. معیارهای مبتنی بر نحو (۲ هفته)
۳. ملاحظات عملی آزمون نرم افزار (۲ هفته)
 - a. آزمون رگرسیونی
 - b. تجمیع و آزمون
 - c. آزمون پذیرش
 - d. آزمون آلفا و بتا
 - e. فرایند آزمون
 - f. طرح آزمون
 - g. تولید آزمون رانه

- h. آزمون نرم‌افزارهای تحت وب
- i. تعیین خروجی صحیح
- j. گزارش اشتباهات، ردیابی و تجزیه و تحلیل نتایج
- ۴. ابزارهای آزمون و تنظیم نرم‌افزار برای آزمون (۱ هفته)
 - a. تنظیم برنامه برای آزمون مبتنی بر گراف و منطق
 - b. ابزارهای آزمون موتاسیونی

آزمون - تمرین - پروژه

- دانشجویان به گروه‌های ۲ یا ۳ نفره تقسیم می‌شوند و هر گروه سه صورت برنامه در اندازه‌های کوچک، متوسط و بزرگ را پیشنهاد می‌نماید. پس از تصویب برنامه‌ها، هر گروه تمرینات (حداقل ۳ تمرین) را در طول ترم براساس مسایل پیشنهادی خود پاسخ خواهد داد.
- برنامه بزرگ صورت پروژه هر گروه را مشخص می‌کند که یک ماه پس از پایان امتحانات فرصت دارند تا پروژه خود را تحویل نمایند. دانشجویان باید یا حتی‌الامکان با استفاده از نرم‌افزارهای موجود در این زمینه کار نمایند.
- امتحان میان ترم
- امتحان پایان ترم

مراجع اصلی

- P. Ammann, J. Offutt, *Introduction to Software Testing*, Cambridge University Press, 2008.
- R.S. Pressman, *Software Engineering: A Practitioner's Approach*, 7th edition, McGraw-Hill, 2010.



Software Engineering 2

طراح درس: سید حسن میریان حسین آبادی

گرایش: نرم افزار	مقطع: تحصیلات تکمیلی
واحد: ۳	نوع درس: اجباری (زیرگرایش مهندسی نرم افزار)
پیش نیاز: -	هم نیاز: -

کلیات

این درس برای دانشجویان کارشناسی ارشد و دکتری ارائه می شود و هدف از آن پرداختن به مباحث پیشرفته در ارتباط با مهندسی نرم افزار است. در این درس روش های جدید در مورد هر یک از مراحل چرخه حیات نرم افزار مورد بحث قرار می گیرند که از آن جمله می توان به استفاده از روش های صورتی (جبری) در ثبت نیازها، روش خط محصول در تجزیه و تحلیل و طراحی و پیاده سازی، مهندسی امنیت، روش جنبه گرا، تولید مبتنی بر آزمون، و روش سرویس گرا اشاره کرد. ابزارهای لازم برای به کارگیری این روش ها به اختصار و در حد امکان معرفی می گردند.

ریز مواد

۱. یادآوری: مروری بر متدولوژی های چرخه حیات، مدیریت پروژه، برآورد، مدیریت ریسک، اندازه گیری و آزمون
۲. روش های صورتی ثبت نیازها و زبان های جبری
۳. متدولوژی خط محصول
۴. روش جنبه گرا
۵. تولید مبتنی بر آزمون
۶. روش سرویس گرا

آزمون - تمرین

- حدود ۵ تمرین نظری و عملی (۲۰ نمره)
- آزمون های کلاسی (۲۰ نمره)
- آزمون نهایی (۶۰ نمره)

مراجع اصلی

- R.S. Pressman, *Software Engineering: A Practitioner's Approach*, 7th edition, McGraw-Hill, 2010.
- I. Sommerville, *Software Engineering*, 7th edition, Addison-Wesley, 2004.
- D. Weiss, C.T.R. Lai, *Software Product-Line Engineering: A Family-Based Software Development Process*, Addison-Wesley, 1999.
- K. Pohl, G. Bockle, F. van der Linden, *Software Product-Line Engineering: Foundations, Principles, Techniques*, Springer, 2005.



طراحی پایگاه داده‌ها ۲ (۴۰۹۳۸)

Database Design 2

طراح درس: سید محمد تقی روحانی رانکوهی

گرایش: نرم افزار	مقطع: تحصیلات تکمیلی
واحد: ۳	نوع درس: اختیاری (زیرگرایش مهندسی نرم افزار)
پیش نیاز: -	هم نیاز: -

کلیات

در این درس با مفاهیم پیشرفته طراحی پایگاه داده‌ها آشنا می‌شویم و در انتهای ترم انتظار می‌رود که دانشجو بر مفاهیمی چون تراکنش، همروندی و پروتکل‌های کنترل آن، ترمیم و ایمنی پایگاه داده‌ها، بهینه‌سازی و دیگر مواردی که در یخس ریزموارد به تفصیل بیان شده‌اند، تسلط یابد.

ریز مواد

۱. تراکنش

تعریف، خواص، حالات، تکنیک نقطه نگهداشت، زیرسیستم مدیریت تراکنش‌ها

۲. مفاهیم تئوری توالی‌پذیری

طرح اجرای متوالی، طرح اجرای همروندی، مشکلات توارد کنترل‌نشده، طرح توالی‌پذیر، طرح‌های معادل (نتیجه‌ای-تعارضی-نمایی)، انواع طرح‌های توالی‌پذیر (نتیجه‌ای-تعارضی-نمایی)، آزمون توالی‌پذیری، قضیه بنیادی در تئوری توالی‌پذیری، کاربرد تئوری توالی‌پذیر

۳. پروتکل‌های کنترل همروندی

پروتکل‌های قفل‌گذاری: دوگانی، چند اسلوبی، قفل‌گذاری دومرحله‌ای (مبنایی-محافظه‌کار-شدید-جسور-دقیق)، پروتکل قفل‌گذاری روی چند واحد قفل‌شدنی، قفل‌گذاری قصدی، قفل‌گذاری درختی، قفل‌گذاری جنگلی، قفل‌گذاری چند نسخه‌ای، راه‌حل‌های مشکل بن‌بست (پیش‌گیری-اجتناب-کشف)، پروتکل‌های مبتنی بر زمان مهر، پروتکل‌های چند نسخه‌سازی، پروتکل تایید، پدیده شبح داده، قفل‌گذاری مسندی، کنترل همروندی در شاخص درختی، پارامترهای ارزیابی تکنیک‌های کنترل همروندی

۴. ترمیم پایگاه داده‌ها

تعریف، انواع خرابی، مدیر ترمیم، روش‌های تخلیه حافظه نهان، امکانات ترمیم، ثبت با نوشتن پیش‌رس، انواع فایل ثبت، زدایش فایل ثبت، ایجاد نقطه واریسی، روش‌های ترمیم خرابی سیستمی (UNDO-NO REDO, REDO-NO UNDO, REDO-UNDO, NO UNDO-NO REDO)، ترمیم خرابی رسانه‌ای

۵. جامعیت داده‌ها

تعریف، عوامل نقض جامعیت، انواع محدودیت جامعیتی، روش‌های توصیف محدودیت جامعیتی، سیستم فعال، قاعده فعال، معماری سیستم فعال، مدل اجرا، مراحل اعمال محدودیت‌ها توسط سیستم فعال، مزایا و کاربردهای سیستم فعال، رهانا (تعریف-کاربرد-مزایا-مشکلات-کم‌داشت‌ها)

۶. ایمنی پایگاه داده‌ها

تعریف، خطرات، تحلیل خطرات، شیء ایمنی، تدابیر ایمنی غیر کامپیوتری، تدابیر کامپیوتری: شناسایی کاربر، احراز هویت، مجازشماری، روش‌های کنترل دستیابی، روش‌های طراحی سیستم اطلاعاتی ایمن

۷. بهینه‌سازی پرسش

تعریف- مراحل کلی پردازش پرسش- تجزیه پرسش- پیاده‌سازی عملگرهای جبر رابطه‌ای، بهینه‌سازی پرسش: بازنویسی، ارزیابی عبارات، انتخاب طرح اجرا: روش مبتنی بر هزینه، روش یافتاری، روش‌های دیگر

۸. پارامترهای شناخت سیستم مدیریت پایگاه داده‌ها

رده‌بندی سیستم، اجزای سیستم، کاتالوگ سیستم، پارامترهای مربوط به کارایی، پارامترهای مربوط به معماری، پارامترهای مربوط به زبان داده‌ای فرعی، محورهای اصلی مقایسه سیستم‌ها، محک‌زنی سیستم

آزمون - تمرین - گزارش پژوهشی - پروژه

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

مراجع اصلی

- R. Elmasri, S.B. Navathe, *Fundamentals of database systems*, 4th edition, Addison-Wesley, 2003.
- A. Silberschatz, H.F. Korth, S. Sudarshan, *Database Systems Concepts*. 3rd edition, McGraw-Hill, 1997.
- C.J. Date, *An Introduction to Database Systems, Volume 2*. Addison-Wesley, 1983.
- H. Garcia-Molina, J.D. Ullman, J. Widom, *Database System Implementation*. Prentice Hall, 2000.
- G. Weikum, G. Vossen, *Transactional Information Systems*. Morgan Kaufmann Publishers, 2002.
- P.A. Bernstein, V. Hadzilacos, N. Goodman, *Concurrency Control and Recovery in Database Systems*. Addison-Wesley, 1987.
- T.M. Connolly, C.E. Begg, *Database Solutions: A step by step guide to building databases*. Addison Wesley, 2003.
- D.A. Simovici, R.L. Tenney, *Relational Database Systems*, Academic Press, 1995.
- J. Gray, A. Reuter, *Transaction Processing: Concepts and Techniques*. Morgan Kaufmann, 1993.

• روحانی رانکوهی: سیستم مدیریت پایگاه داده‌ها: مفاهیم و تکنیک‌ها...

• بازخوانی درس "طراحی و پیاده‌سازی پایگاه داده‌ها" (کارشناسی):

○ منابع ۱ تا ۳ و ۷ تا ۹

○ روحانی رانکوهی، مفاهیم بنیادی پایگاه داده‌ها، از چاپ سوم به بعد.



مهندسی نیازمندی‌ها (۴۰---)

Requirements Engineering

طراح درس: عباس حیدرنوری

گرایش: نرم‌افزار	مقطع: تحصیلات تکمیلی
واحد: ۳	نوع درس: اجباری (زیرگرایش مهندسی نرم‌افزار)
پیش‌نیاز: -	هم‌نیاز: -

کلیات

پیاده‌سازی یک سیستم کامپیوتری با شکست مواجه خواهد شد اگر به نیازمندی‌های آن سیستم به درستی پاسخ داده نشود. همراه با گسترش و افزایش پیچیدگی سیستم‌های کامپیوتری، شناسایی دقیق نیازمندی‌های آنها نیز به فرایندی پیچیده تبدیل گردیده است. بنابراین، لازم است تا از روش‌ها و تکنیک‌هایی برای شناسایی، توصیف و مستندسازی، مدل‌سازی، و اعتبارسنجی نیازمندی‌های سیستم‌های کامپیوتری استفاده شود. هدف از این درس، آشناسازی دانشجویان مقاطع تحصیلات تکمیلی با این روش‌ها و تکنیک‌هاست. به طور خاص، مباحثی که در این درس مطرح خواهند شد عبارتند از: (۱) استخراج و شناسایی نیازمندی‌های وظیفه‌مندی و غیروظیفه‌مندی، (۲) سازمان‌دهی و اولویت‌بندی نیازمندی‌ها، (۳) تکنیک‌های توصیف و مستندسازی نیازمندی‌ها، (۴) تکنیک‌های مدل‌سازی نیازمندی‌ها، (۵) تکنیک‌های تحلیل، واریسی، و اعتبارسنجی نیازمندی‌ها.

ریز مواد

۱. مبانی مهندسی نیازمندی‌ها
۲. درک دامنه مسئله و استخراج نیازمندی‌ها
۳. ارزیابی نیازمندی‌ها
۴. توصیف و مستندسازی نیازمندی‌ها
۵. تضمین کیفی نیازمندی‌ها
۶. تکامل نیازمندی‌ها و ردیابی
۷. مقصودگرایی در مهندسی نیازمندی‌ها
۸. مدل‌سازی اهداف سیستم با استفاده از مدل‌های مقصود
۹. تحلیل خطر با استفاده از مدل‌های مقصود
۱۰. مدل‌سازی نیازمندی‌ها با استفاده از نمودارهای سناریوگرا
۱۱. مدل‌سازی نیازمندی‌ها با استفاده از نمودارهای UML
۱۲. مدل‌سازی عملیات سیستم
۱۳. مدل‌سازی رفتار سیستم
۱۴. واریسی و اعتبارسنجی نیازمندی‌ها
۱۵. مدیریت نیازمندی‌ها

آزمون - ارائه مقاله علمی - پروژه پژوهشی

- آزمون میان ترم: ۲۵٪ کل نمره
- آزمون پایان ترم: ۳۵٪ کل نمره
- ارائه مقاله علمی: هر دانشجو موظف به ارائه یک مقاله علمی می‌باشد که در یکی از بهترین کنفرانس‌ها یا مجلات مرتبط به درس در سال‌های اخیر به چاپ رسیده باشد (۱۰٪ کل نمره).
- پروژه و گزارش پژوهشی: موضوع پروژه پژوهشی قبل از آزمون نیمسال تعیین می‌شود. دانشجو کار پژوهش را با کمک استاد درس آغاز نموده و پس از انجام پروژه، نتیجه پژوهش را در قالب گزارش ارائه می‌دهد (۳۰٪ کل نمره).

مراجع اصلی

- A. van Lamsweerde, *Requirements Engineering: From System Goals to UML Models to Software Specifications*, Wiley, 2009.
- K. Pohl, *Requirements Engineering: Fundamentals, Principles, and Techniques*, Springer, 2010.

مراجع فرعی

- G. Kotonya, I. Sommerville, *Requirements Engineering: Processes and Techniques*, Wiley, 1998.
- S. Robertson, J.C. Robertson, *Mastering the Requirements Process*, 2nd Edition, Addison-Wesley Professional, 2006.
- B. Berenbach, D. Paulish, J. Kazmeier, A. Rudorfer, *Software & Systems Requirements Engineering: In Practice*, McGraw-Hill, 2009.
- E. Hull, K. Jackson, J. Dick, *Requirements Engineering*, 3rd Edition, Springer, 2010.
- J. Holt, S. Perry, M. Brownsword, *Model-Based Requirements Engineering*, Institution of Engineering and Technology (IET), 2011.
- D. Leffingwell, *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*, Addison-Wesley Professional, 2011.
- M. Chemuturi, *Requirements Engineering and Management for Software Development Projects*, Springer, 2012.



تکامل نرم افزار (۴۰---)

Software Evolution

طراح درس: عباس حیدرنوری

گرایش: نرم افزار	مقطع: تحصیلات تکمیلی
واحد: ۳	نوع درس: اجباری (زیرگرایش مهندسی نرم افزار)
پیش نیاز: -	هم نیاز: -

کلیات

تولیدکنندگان نرم افزار اغلب به جای تولید سیستم های نرم افزاری جدید، حجم زیادی از منابع خود را صرف مراقبت و نگهداری از سیستم های موجود می کنند. این مراقبت و نگهداری اغلب شامل بهبود طراحی و پیاده سازی سیستم، پیدا کردن و رفع خطاها، و افزودن امکانات جدید به نرم افزار می باشد. هدف از فرایند "تکامل نرم افزار"، حصول اطمینان از ادامه کار موفقیت آمیز یک سیستم نرم افزاری بعد از مرحله تولید آن می باشد. ولیکن، این فرایند، به خودی خود، یک فرایند پیچیده و زمان بر می باشد به گونه ای که روش ها و ابزارهای گوناگونی برای این منظور طراحی و پیاده سازی گردیده است. هدف این درس، آشنا کردن دانشجویان کارشناسی ارشد و دکترا با مفاهیم "تکامل نرم افزار"، و آخرین پژوهش ها و ابزارهای انجام شده در این زمینه می باشد. این درس، یکی از دروس مطرح در دوره تحصیلات تکمیلی مهندسی نرم افزار در چندین دانشگاه معتبر دنیا می باشد. برخی از این دانشگاه ها عبارتند از: McGill University (شماره درس: COMP762)، University of Waterloo (شماره درس: CS846)، Kent State University (شماره درس: CS6/73902)، و University of Texas at Austin (شماره درس: EE382V).

ریز مواد

- بخش اول: مفاهیم مقدماتی تکامل نرم افزار
 - آشنایی با تکامل نرم افزار و مفاهیم اولیه (۱ جلسه - مدت هر جلسه ۹۰ دقیقه می باشد).
 - تحلیل ایستا در مقابل تحلیل پویای سیستم های نرم افزاری (۱ جلسه)
- بخش دوم: مهندسی معکوس سیستم های نرم افزاری
 - مصورسازی سیستم های نرم افزاری (۲ جلسه)
 - تحلیل و شناسایی کلونی های نرم افزاری (۲ جلسه)
 - جستجو در کد برنامه به منظور شناسایی نحوه پیاده سازی یک امکان خاص (۲ جلسه)
 - برنامه نویسی جنبه گرا و جستجو در سیستم های نرم افزاری به منظور شناسایی جنبه ها (۲ جلسه)
 - شناسایی خطاها (۲ جلسه)
 - داده کاوی در مخزن های نرم افزاری (۲ جلسه)
 - درک و تحلیل چهارچوب های نرم افزاری (۲ جلسه)
- بخش سوم: بازمهندسی سیستم های قدیمی
 - مزایای پیمانه ای کردن نرم افزار (۱ جلسه)
 - بازمهندسی شیء گرا (۱ جلسه)
 - به روزرسانی سیستم های قدیمی (۱ جلسه)
 - بهبود و بازآرایی طراحی (۱ جلسه)

- بخش چهارم: مفاهیم جدید در تکامل نرم افزار
 - سیستم‌های توصیه‌گر (۲ جلسه)
 - تکامل API (۲ جلسه)
 - رابطه بین تکامل نرم افزار و آزمون نرم افزار (۲ جلسه)
 - تکامل معماری نرم افزار (۲ جلسه)
 - مطالعه تجربی تکامل نرم افزار (۲ جلسه)

آزمون - ارائه مقاله علمی - پروژه پژوهشی

- آزمون میان ترم: ۲۰٪ کل نمره
- آزمون پایان ترم: ۳۵٪ کل نمره
- ارائه مقاله علمی: هر دانشجو موظف به ارائه حداقل یک مقاله علمی می‌باشد که در یکی از بهترین کنفرانس‌ها یا مجلات مرتبط به درس در سال‌های اخیر به چاپ رسیده باشد (۱۵٪ کل نمره).
- پروژه و گزارش پژوهشی: موضوع پروژه پژوهشی قبل از آزمون نیمسال تعیین می‌شود. دانشجو کار پژوهش را با کمک استاد درس آغاز نموده و پس از انجام پروژه، نتیجه پژوهش را در قالب گزارش ارائه می‌دهد (۳۰٪ کل نمره).

مراجع اصلی

- T. Mens, S. Demeyer (Eds.), *Software Evolution*, Springer, 2008.
- S. Jarzabek, *Effective Software Maintenance and Evolution: A Reuse-Based Approach*, Auerbach Publications, 2007.
- N.H. Madhavji, J. Fernandez-Ramil, D. Perry (Eds.), *Software Evolution and Feedback: Theory and Practice*, Wiley, 2006.
- A. Zeller, *Why Programs Fail: A Guide to Systematic Debugging*, Morgan Kaufmann, 2009.
- S. Diehl, *Software Visualization: Visualizing the Structure, Behaviour, and Evolution of Software*, Springer, 2010.
- P. Tonella, A. Potrich, *Reverse Engineering of Object Oriented Code*, Springer, 2005.
- M. Lippert, S. Roock, *Refactoring in Large Software Projects: Performing Complex Restructurings Successfully*, Wiley, 2006.
- M. Fowler, K. Beck, J. Brant, W. Opdyke, D. Roberts, *Refactoring: Improving the Design of Existing Code*, Addison-Wesley Professional, 1999.

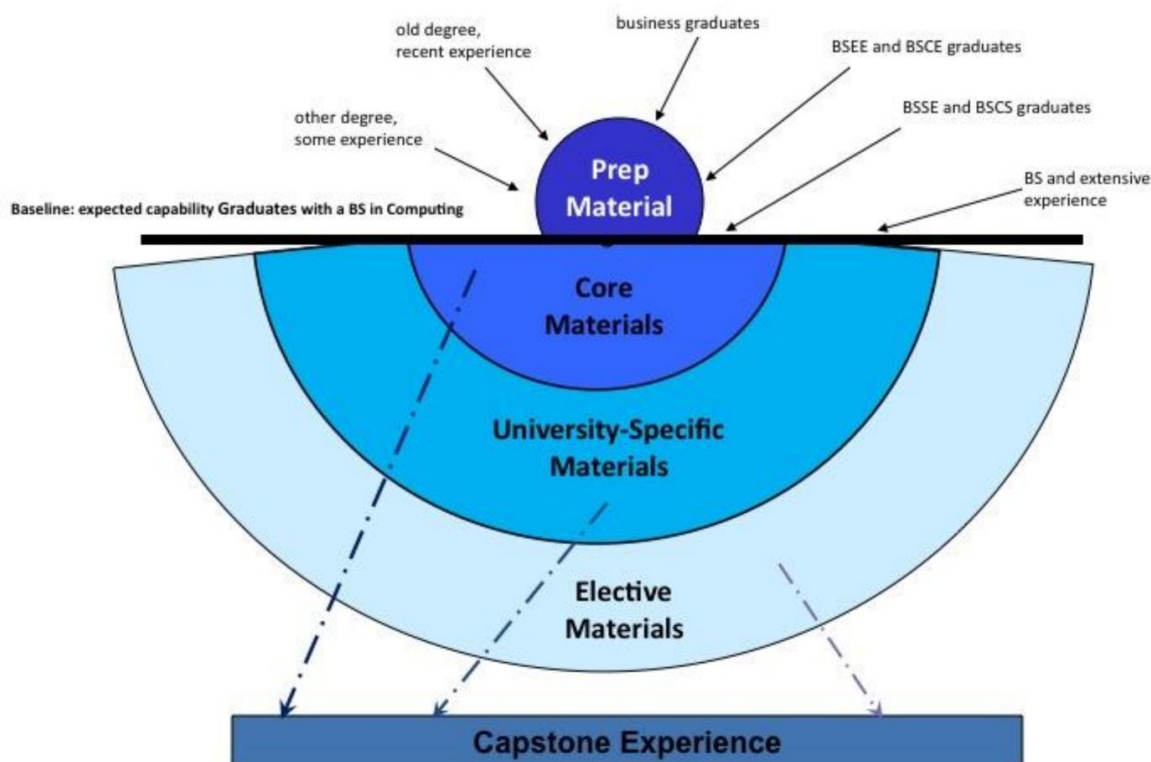
ضمائم

ضمیمه الف: ساختار کلی برنامه پیشنهادی ACM/IEEE

برای کارشناسی ارشد مهندسی نرم افزار

معماری قابل پیکربندی ACM/IEEE برای دوره کارشناسی ارشد مهندسی نرم افزار (نسخه ۲۰۰۹[‡]) شامل اجزاء ساختاری زیر است (طبق شکل ۱):

۱. مواد درسی جبرانی (Prep Materials): جهت آماده سازی دانشجویان پذیرفته شده برای آغاز دوره.
۲. مواد درسی اصلی (Core Materials): حداکثر نیمی از واحدهای درسی دوره باید به مباحث بدنه اصلی دانش مهندسی نرم افزار - CBOK[§] - تخصیص یابد؛ حوزه های دانش CBOK به تفصیل در ضمیمه ب توضیح داده شده اند.
۳. مواد درسی خاص دانشگاه (University-Specific Materials): این واحدهای درسی مطابق با اهدافی که دانشگاه از ارائه دوره در نظر دارد تعریف می شوند.
۴. مواد درسی اختیاری (Elective Materials): این واحدهای درسی به دانشجو امکان می دهند تا دوره را طبق علائق تخصصی خود جهت دهی نماید.
۵. تجربه پایانی (Capstone Experience): در قالب پروژه یا رساله.



شکل ۱. معماری کلی برنامه پیشنهادی ACM/IEEE برای کارشناسی ارشد مهندسی نرم افزار

[‡] این برنامه در سایت اینترنتی روبرو موجود است: <http://www.gswe2009.org/curriculum/recommendations>

[§] Core Body of Knowledge

ضمیمه ب: مباحث بدنه اصلی دانش مهندسی نرم افزار – CBOK

بدنه اصلی دانش مهندسی نرم افزار (CBOK)**، طبق برنامه پیشنهادی ACM/IEEE برای کارشناسی ارشد مهندسی نرم افزار (نسخه ۲۰۰۹^{††})، شامل ۱۱ حوزه کلی دانش است که در زیر فهرست شده‌اند:

- A. آداب و رفتار حرفه‌ای (Ethics and Professional Conduct)
- B. مهندسی سیستم (System Engineering)
- C. مهندسی خواسته‌ها (Requirements Engineering)
- D. طراحی نرم افزار (Software Design)
- E. ساخت نرم افزار (Software Construction)
- F. آزمون (Testing)
- G. مراقبت و نگهداری از نرم افزار (Software Maintenance)
- H. مدیریت پیکربندی (Configuration Management)
- I. مدیریت مهندسی نرم افزار (Software Engineering Management)
- J. فرایند مهندسی نرم افزار (Software Engineering Process)
- K. کیفیت نرم افزار (Software Quality)

هریک از حوزه‌های دانش فوق شامل مباحث ریزدانه‌تری است که در جدول ب آورده شده‌اند.

جدول ب. مباحث ریزدانه بدنه اصلی دانش مهندسی نرم افزار (CBOK)

A. Ethics and Professional Conduct	1. Social, legal, and historical issues	Data confidentiality and security, surveillance and privacy Historical developments, and gender, minor, and cultural issues Contracts and liability, intellectual property, freedom of information Computer crime and law enforcement
	2. Codes of ethics and professional conduct	Responsibilities to society Models for professionalism, professional societies Codes of ethics and practice
	3. The nature and role of software engineering standards	Nature and role of standards International standards, standards and harmonization organizations Bodies of knowledge, accepted and best practices
B. System Engineering	1. Systems Engineering Concepts	System context People and systems System hierarchical relationships The role of system engineers
	2. System Engineering Life Cycle Management	Lifecycle Management Systems engineering and software engineering processes
	3. Requirements	Stakeholder requirements Requirements analysis
	4. System Design	Architectural design Implementation Trade studies
	5. Integration and Verification	
	6. Transition and Validation	
	7. Operation, Maintenance and Support	

** Core Body of Knowledge

^{††} این برنامه در سایت اینترنتی روبرو موجود است: <http://www.gswe2009.org/curriculum/recommendations>

C. Requirements Engineering	1. Fundamentals of Requirements Engineering	Relationship between systems engineering and software engineering Definition of requirements System design constraints System design and requirements allocation Product and process requirements Functional and non-functional requirements Emergent properties Quantifiable requirements
	2. Requirements Engineering Process	Process models Process actors Process support and management Process quality and improvement
	3. Initiation and Scope Definition	Determination and negotiation of requirements Feasibility analysis Process for requirements review/revision
	4. Requirements Elicitation	Requirements sources Elicitation techniques
	5. Requirements Analysis	Requirements classification Conceptual modeling Heuristic methods Formal methods Requirements negotiation
	6. Requirements Specification	Requirements specification techniques
	7. Requirements Validation	Requirements reviews Prototyping Model validation Acceptance tests
	8. Practical Considerations	Iterative nature of requirements process Change management Requirements attributes Requirements tracing Measuring requirements
D. Software Design	1. Software Design Fundamentals	General design concepts Context of software design Software design process Enabling techniques
	2. Key Issues in Software Design	Concurrency Control and handling of events Distribution of components Error and exception handling and fault tolerance Interaction and presentation Data persistence
	3. Software Structure and Architecture	Architectural structures and viewpoints Architectural styles (macro architectural patterns) Design patterns (micro architectural patterns) Human computer interface design Families of programs and frameworks
	4. Software Design Quality Analysis and	Evaluation Quality attributes Quality analysis and evaluation techniques Measures
	5. Software Design Notations	Structural descriptions (static) Behavioral descriptions (dynamic)
	6. Software Design Strategies and Methods	General strategies Function-oriented (structured) design Object-oriented design Heuristic methods Formal methods Component-based design (CBD)
E. Software Construction	1. Software Construction Fundamentals	Minimizing complexity Anticipating change Constructing for verification Standards in construction

	2. Managing Construction	Construction methods Construction planning Construction measurement
	3. Practical Considerations	Construction design Coding Construction testing Construction quality Integration
F. Testing	1. Testing Fundamentals	System testing and software testing Testing-related terminology Key issues Relationships of testing to other activities
	2. Test Levels	The target of the tests Objectives of testing Component testing Integration testing System testing Acceptance testing
	3. Testing Techniques	Based on tester's intuition and experience Specification-based Code-based Fault-based Usage-based Based on nature of application Selecting and combining techniques
	4. Test-Related Measures	Evaluation of the program or system under test Evaluation of the tests performed
	5. Test process	Management concerns Test activities
G. Software Maintenance	1. Software Maintenance Fundamentals	Definitions and terminology Nature of maintenance Need for maintenance Majority of maintenance costs Evolution of software Categories of maintenance
	2. Key Issues in Software Maintenance	Technical - Limited understanding - Testing - Impact analysis - Maintainability Management issues - Alignment with organizational issues Maintenance cost estimation - Cost estimation - Parametric models - Experience Software maintenance measurement
	3. Maintenance Process	Maintenance process models Maintenance activities - Unique activities - Supporting activities
	4. Techniques for Maintenance	Program comprehension Reengineering Reverse engineering
H. Configuration Management (CM)	1. Management of the CM Process	Organizational context for CM Constraints and guidance for CM Planning for CM - CM organization and responsibilities - CM resources and schedules - Vendor/subcontractor control - Interface control Configuration management plan Surveillance of configuration management - CM measures and measurement - In-process audits of CM
۳۶		

	2. Configuration Identification	Identifying items to be controlled - Configuration items - Configuration item relationships - Versions - Baseline - Acquiring configuration items Software library
	3. Configuration Control	Requesting, evaluating and approving changes - Configuration control board - Change request process Implementing changes Deviations and waivers
	4. Configuration Status Accounting	Configuration status reporting
	5. Software Release Management and Delivery	Software building Software release management
I. Software Engineering Management	1. Software Project Planning	Project goals and objectives Project policies and standards Process planning Project assumptions and forecasts Project deliverables Project staffing Effort, schedule, and cost estimation Resource allocation Quality management Project plan/budget development and management
	2. Risk Management	Risk management concepts - Probability, impact - Timeframe Risk management process - Frameworks, standards, and guidelines - Risk identification, analysis and risk prioritization techniques - Risk mitigation strategies Risk management tools - Earned value tracking - Technical performance measurement - Defect tracking and reporting - Project control panels Organizational risk management Joint supplier/customer risk management
	3. Software Project Organization and Enactment	Project organization - Identify and group project functions, activities, and tasks - Determine organizational structure and positions - Define responsibilities, authority relationships, position qualifications Project directing - Leadership, supervision, delegation of authority, coordination and communication - Motivation, conflict resolution, team building Project control - Implementation of plans, and measurement process - Process monitoring - Change management Reporting Supplier contract management (e.g., RFP, cost evaluation, IP rights)
	4. Review and Evaluation	Determining satisfaction of requirements Reviewing and evaluating performance
	5. Closure	Determining closure Closure activities
	6. Software Engineering Measurement	Establish and sustain measurement commitment Plan the measurement process Perform the measurement process Evaluate measurement

	7. Engineering Economics	Engineering economics fundamentals For-profit decision-making Not-for-profit decision-making Present economy Estimation, risk, and uncertainty Multiple attribute decisions
J. Software Engineering Process	1. Process Implementation and Change	Process infrastructure - Software engineering process group - Experience factory Activities Models for process implementation and change Practical considerations
	2. Process Definition	Life cycle models Software life cycle processes Notations for process definitions Process adaptation Automation
	3. Process Assessment	Process assessment models Process assessment methods
	4. Product and Process Measurement	Software process measurement Software product measurement - Size measurement - Structure measurement - Quality measurement Quality of measurement results Measurement techniques - Analytical techniques - Benchmarking techniques
K. Software Quality	1. Software Quality Fundamentals	Software engineering culture and ethics Value and costs of quality Quality models and characteristics - Software process quality - Software product quality Quality improvement Application quality requirements - Criticality of systems - Dependability - Integrity levels of software Defect characterization
	2. Software Quality Management Processes	Software quality assurance Software quality management techniques - Static techniques - People-intensive techniques - Analytic techniques - Dynamic techniques Software quality measurement
	3. Verification and Validation (V&V)	Definitions of V&V - System V&V and software V&V - Independent V&V V&V Techniques - Testing - Demonstrations - Traceability - Analysis - Inspections - Peer reviews - Walkthroughs - Audits