

فارسی: اصول طراحی کامپایلر		تعداد واحد: نظری ۳		مقطع: <input checked="" type="checkbox"/> کارشناسی <input type="checkbox"/> کارشناسی ارشد <input type="checkbox"/> دکتری		
نام درس		لاتین: Fundamentals of Compiler Design		پیش نیاز: نظریه زبان ها و ماشین ها		
مدرس: مرتضی درّی گیو		شماره تلفن دفتر کار (اتاق ۳۷۳): ۰۲۳-۳۱۵۳۲۷۰۸				
پست الکترونیکی: dorrigiv@semnan.ac.ir		منزلگاه اینترنتی: http://dorrigiv.profile.semnan.ac.ir				
برنامه تدریس در هفته: شنبه‌ها (ساعت ۱۵ تا ۱۷ - مجازی) و سه‌شنبه‌ها (ساعت ۱۳ تا ۱۴ - مجازی)						
<p>اهداف درس: مطالعه درباره‌ی طراحی و ساخت کامپایلرها یکی از مفاهیم بنیادی علوم کامپیوتر است و به دانشی در خصوص هم زبان منبع و هم زبان مقصد نیاز دارد و پیوندی مابین این دو برقرار می‌کند. امروزه نرم‌افزارهایی نیز موجود است که می‌تواند ساخت کامپایلرها را تسهیل کند. بیشتر مباحث نظری و بسیاری از روش‌های ساخت کامپایلرها از دهه ۱۹۷۰ و بعضاً حتی پیش از آن پدید آمده است. اما از طرفی توسعه‌ی زبان‌های برنامه‌سازی جدید، یک مبارزه‌ی دائمی برای کامپایلر نویسان بوجود می‌آورد. نیاز به مبانی و تکنیک‌های پیاده‌سازی کامپایلرها آن چنان در حال گسترش است که ایده‌های مطرح شده در این درس، ممکن است به کرات مورد استفاده متخصصان کامپیوتر واقع شود. طراحی و ساخت کامپایلرها با مفاهیم زبان‌های برنامه‌سازی، نظریه زبان‌ها و ماشین‌ها، معماری کامپیوتر، طراحی الگوریتم‌ها و مهندسی نرم‌افزار مرتبط است. خوشبختانه روش‌های ساخت کامپایلرها علی‌رغم آن‌که تنوع کمی دارند، لیکن می‌توانند برای ساخت مترجم‌های طیف گسترده و متنوعی از زبان‌ها و ماشین‌ها استفاده شوند. در این درس، موضوع ساخت کامپایلرها از طریق توصیف مؤلفه‌های اصلی یک کامپایلر و محیطی که یک کامپایلر در آن کار می‌کند، معرفی می‌شود. پس از معرفی مقدماتی درباره‌ی اجزاء یک کامپایلر و انواع گرامرها، مراحل مختلف ترجمه از قبیل تجزیه و تحلیل لغوی، نحوی و معنایی و غیره تشریح خواهد شد. در پایان، به معرفی تکنیک‌های بهینه‌سازی کد و خطاپردازی پرداخته می‌شود.</p>						
زمان امتحان: میان‌ترم اول (۸ آبان ۱۴۰۰ - ساعت ۱۵)، میان‌ترم دوم (۲۰ آذر ۱۴۰۰ - ساعت ۱۵) و پایان‌ترم (۲۸ دی ۱۴۰۰ - ساعت ۱۶)						
نحوه ارزشیابی	تمرین (E)	امتحان میان‌ترم اول (M ₁)	امتحان میان‌ترم دوم (M ₂)	امتحان پایان‌ترم (F)	فعالیت کلاسی (Q)	رشد (P)
درصد نمره	۱۵	۲۵	۲۵	۲۵	۱۰	اضافی
فرمول محاسبه نمره	$P = \text{if } M_2 > M_1 \text{ then } (M_2 > M_1) / 2 \text{ else } 0$ $G = (E + M_1 + M_2 + F + Q + P) / 5$					
قوانین درس	<p>۱- تحویل به موقع تمرین‌ها در سامانه امید است.</p> <ul style="list-style-type: none"> تاریخ تحویل تمرین‌ها در صورت موافقت استاد، فقط و فقط برای یک تمرین با این شرط که دو روز قبل از مهلت تحویل دو سوم کلاس درخواست تمدید داشته باشند، تمدید خواهد شد. برای هر تمرین به ازای هر روز تأخیر ۲۰٪ از نمره اخذ شده‌ی آن کسر خواهد شد. قانون محاسبه‌ی تأخیر تنها تا زمان حل تمرین‌ها در کلاس حل تمرین برقرار خواهد بود و در صورت تحویل بعد از حل تمرین نمره‌ای در نظر گرفته نمی‌شود. تمرین‌ها می‌توانند در قالب گروه‌های یک یا دو نفره تحویل گردند. تفاوتی بین گروه‌های یک یا دو نفره وجود نخواهد داشت. <p>۲- صفحه درس در سامانه امید برای تمام پرسش و پاسخ‌ها در نظر گرفته شده است، بنابراین لطفاً همواره این صفحه را پیگیری کنید.</p> <p>۳- لازم به تذکر است که تشخیص تقلب و یا کپی مستقیم از منبعی، بنا به تشخیص تصحیح‌کننده، باعث صفر شدن آن تمرین می‌شود.</p> <p>۴- در صورتی که نمره میان‌ترم دوم نسبت به نمره میان‌ترم اول بهبود داشته باشد، نمره رشد به دانشجو تعلق می‌گیرد.</p> <p>۵- حضور در کلاس درس الزامی نیست. ولی نمره‌های اضافی تنها به دانشجویانی تعلق می‌گیرد که بیش از نیمی از جلسه‌های درس را حضور داشته باشند.</p>					
منابع و مآخذ درس	<p>مراجع اصلی:</p> <p>Torben Aegidius Mogensen, "Introduction to Compiler Design," Springer International Publishing, Second Edition, 2017.</p> <p>Alfred V. Aho, Monica S. Lam, Ravi Sethi, and Jeffrey D. Ullman, "Compilers : principles, techniques, and tools (dragonbook)," Addison-Wesley, Second Edition, 2007.</p> <p>مراجع دیگر: به صفحه درس در سامانه امید مراجعه شود.</p>					

[نیمسال اول ۰۱-۰۰]، [نیمسال دوم ۰۰-۹۹]، [نیمسال اول ۰۰-۹۹]، [نیمسال دوم ۹۹-۹۸]، [نیمسال اول ۹۸-۹۹]، [نیمسال دوم ۹۸-۹۷]، [نیمسال اول ۹۸-۹۷]، [نیمسال دوم ۹۷-۹۶]، [نیمسال اول ۹۷-۹۶]، [نیمسال دوم ۹۶-۹۵]، [نیمسال اول ۹۵-۹۶]، [نیمسال دوم ۹۴-۹۵]، [نیمسال اول ۹۴-۹۵]، [نیمسال دوم ۹۳-۹۴]، [نیمسال اول ۹۳-۹۲]، [نیمسال اول ۹۲-۹۱]، [نیمسال اول ۹۱-۹۰]، [نیمسال اول ۹۰-۸۹].	نیمسال‌های ارائه درس
این نیمسال تنها از سامانه امید استفاده می‌شود.	

بودجه‌بندی درس

توضیحات	مبحث	شماره هفته آموزشی
کامپایلر به عنوان یک برنامه سیستمی	تعریف کامپایلر و معرفی جایگاه آن در یک سیستم کامپیوتری	۱
سرعت کامپایلر، سرعت اجرا، و خطاپردازی	تفاوت کامپایلرها و مفسرها	۲
واژه‌یاب، ساختاریاب، کدساز، جدول نمادها، پشته‌ها، بهینه‌ساز و خطاپرداز	تشریح معماری یک کامپایلر	۳
زبان منظم، ماشین حالت و استفاده از Lexer Generatorها	تحلیل لغوی و پیاده‌سازی برنامه واژه‌یاب	۴
زبان مستقل از متن، ماشین اتومات پشته‌ای و Parser Generatorها	تحلیل نحوی و پیاده‌سازی برنامه ساختاریاب	۵
واژه‌های مفهومی و روال‌های مفهومی	تحلیل مفهومی و پیاده‌سازی برنامه کدساز	۶
شبیه‌سازی اشتقاق چپ ورودی، پارسر $LL(1)$	معرفی و پیاده‌سازی پارسرهای بالا به پایین	۷
فاکتورگیری و رفع چپ‌گردی به همراه استفاده تعددی از گرامرهای مبهم	ایجاد گرامر مناسب برای پارسرهای بالا به پایین	۸
عبارت‌های ریاضی، آرایه‌ها، و دستوراتی مانند if-then-else	تولید کد در پارسرهای بالا به پایین	۹
شبیه‌سازی معکوس اشتقاق راست ورودی، پارسر $LR(0)$ ، $SLR(1)$ ، $LALR(1)$ و $LR(1)$	معرفی و پیاده‌سازی پارسرهای پایین به بالا	۱۰
ایجاد گرامر مناسب برای برقراری ارتباط با کدساز	تولید کد در پارسرهای پایین به بالا	۱۱
سطوح بهینه‌سازی، گراف جریان کنترلی، بهینه‌سازی محلی و سراسری	معرفی تکنیک‌های بهینه‌سازی کد	۱۲
تولید کد با پیچیدگی $O(1)$	تولید کد case	۱۳
تکنیک‌های وابسته و ناوابسته به ماشین	بهینه‌سازی محلی	۱۴
خارج نمودن دستورهای ناوابسته به حلقه و بازکردن حلقه	بهینه‌سازی حلقه‌ها	۱۵
خطاهای لغوی، دستوری و مفهومی، خطاهای زمان کامپایلر و زمان اجرا	خطاپردازی	۱۶