به نام ایزد دانا

دانشگاه سمنان

(کامپایلر پیشرفته)

نسخه اولیه: ۱۴۰۰/۶/۲۷

تاریخ به‌روز رسانی: ۱۴۰۰/۶/۳۰

دانشکده مهندسی برق و کامپیوتر

نیمسال اول سال تحصیلی ۱۴۰۱-۱۴۰۰

نام درس	فارسی: کامپایلر پیشرفته	تعداد واحد: نظری ۳	مقطع: کارشناسی☐ کارشناسی ارشد☑ دکتری☐
	لاتین: Advanced Compilers	پیش‌نیاز: اصول طراحی کامپایلر	
مدرس: مرتضی درّی‌گیو		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: دوشنبه‌ها (ساعت ۱۵ تا ۱۷ – مجازی) و سه‌شنبه‌ها (ساعت ۱۵ تا ۱۷/۳۰ – مجازی)

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 آن‌چنان 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. روش‌های استفاده تعمدی از گرامرهای مبهم موضوع دیگری است که در این درس مورد واکاوی قرار می‌گیرد. در پایان، به معرفی تکنیک‌های بهینه‌سازی کد و خطاپردازی پرداخته می‌شود.

زمان امتحان: پایان‌ترم (۲ بهمن ۱۴۰۰– ساعت ۱۰/۳۰)

نحوه ارزشیابی	تمرین (E)	ارائه کار تحقیقی (R)	پیاده‌سازی (I)	امتحان پایان‌ترم (F)	فعالیت کلاسی (Q)	رشد (P)
درصد نمره	۱۵	۲۵	۲۵	۲۵	۱۰	اضافی

فرمول محاسبه نمره: $G = (E + R + I + F + Q + P)/5$

قوانین درس

۱- تحویل به موقع تمرین‌ها در سامانه امید است.
- 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.

۲- صفحه درس در سامانه امید برای تمام پرسش و پاسخ‌ها در نظر گرفته شده است، بنابراین لطفاً همواره این صفحه را پیگیری کنید.
۳- لازم به تذکر است که تشخیص تقلب و یا کپی مستقیم از منبعی، بنا به تشخیص تصحیح‌کننده، باعث صفر شدن آن تمرین می‌شود.
۴- دانشجویان موظف به ارائه کار تحقیقی و پیاده‌سازی هستند.
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.

منابع و مآخذ درس

مراجع اصلی:

Torben Ægidius Mogensen, "Introduction to Compiler Design," *Springer International Publishing*, Second Edition, 2017.

Alfred V. Aho, Monica S. Lam, Ravi Sethi, and Jeffrey D. Ullman, "Compilers : principles, techniques, and tools (dragonbook)," *Addisson-Wesly*, Second Edition, 2007.

مراجع دیگر: به صفحه درس در سامانه امید مراجعه شود.

نیم‌سال‌های ارائه درس	[نیم‌سال اول ۰۱-۰۰]، [نیم‌سال اول ۰۰-۹۹].
صفحه گروه درس	این نیم‌سال تنها از سامانه امید استفاده می‌شود.

بودجه‌بندی درس

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