



طراحی زبان‌های برنامه‌سازی، ۸۱۰۱۴۶۰

Programming Language Design, 8101460										نام انگلیسی درس
واحد: ۳	مهندسی کامپیوتر			مهندسی برق						نوع درس
	فناوری اطلاعات	سخت‌افزار	نرم‌افزار	دیجیتال	کنترل	پزشکی	قدرت	الکترونیک	مخابرات	
	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
										اجباری
										اختیاری
										کارشناسی <input checked="" type="checkbox"/> تحصیلات تکمیلی <input type="checkbox"/>
										مقطع
										ندارد
										همنیازها
										پیش نیازها
										طراحی و ساخت کامپایلرها (۸۱۰۱۴۲۰)
										مطالب پیش نیاز
1. Concepts in Programming Languages by John C. Mitchell, Cambridge University Press, 2004. 2. Foundations for Programming Languages by John C. Mitchell, MIT Press, 2009.										کتاب‌های مرجع
										اهداف درس
اهداف اصلی درس: ۱. بررسی مفاهیم زبان‌های برنامه‌سازی ۲. معرفی ایده‌های زبان‌های برنامه‌سازی به عنوان چارچوبی برای حل مسئله ۳. بررسی روش‌های برنامه‌سازی و امکانات آن‌ها ۴. مقایسه زبان‌های برنامه‌سازی متفاوت ۵. نشان دادن تنوع ایده‌ها در برنامه‌سازی ۶. آماده‌سازی برای کار با زبان‌های برنامه‌سازی جدید، روش‌ها و الگوها ۷. بررسی ویژگی‌های زبان‌های برنامه‌سازی برجسته ۸. مرور trade-off در پیاده‌سازی زبان‌های مختلف ۹. درک هزینه‌های روش‌های انتزاعی										
										نتایج درس
دانشجویانی که این درس را با موفقیت پشت سر بگذارند قادر خواهند بود ۱. معماری زبان‌های برنامه‌نویسی رایج را تجزیه و تحلیل کنند. ۲. با الهام گرفتن از روش‌های طراحی زبان‌های مختلف، ایده‌های جدید ارائه دهند. ۳. ویژگی‌های زبان‌های برنامه‌نویسی برجسته را مقایسه کنند. ۴. دسته‌بندی‌ها و روش‌های مختلف پیاده‌سازی زبان‌های برنامه‌نویسی را بهتر درک کنند. ۵. با روش‌های برنامه‌نویسی جدید بهتر ارتباط برقرار کنند. ۶. از قابلیت‌های یک زبان برنامه‌نویسی در برنامه‌نویسی‌های نرم‌افزاری به صورت کامل استفاده کنند. ۷. رفتار ایستا/پویای یک زبان را تفسیر کنند.										



۸. از ویژگی‌های نوع داده در برنامه‌نویسی به صورت کامل استفاده کنند.	
۹. مفاهیم پیاده‌سازی زبان‌های برنامه‌نویسی مانند جریان کنترل، انتزاع، mutation و غیره را درک کنند.	
۱- قسمت اول: مفاهیم پایه (۲ هفته) (a) مقدمه (b) مفاهیم پایه (c) تاریخچه زبان‌های برنامه‌نویسی (d) تجزیه و تحلیل واژگانی و نحوی	فهرست مباحث
۲- قسمت دوم: میانی و منطق زبان‌های برنامه‌نویسی (۲ هفته) (a) محاسبات لامبدا (b) تئوری نقطه ثابت (c) مقدمه‌ای بر سیستم‌های نوع (d) رسمی‌سازی سیستم‌های نوع (e) سیستم‌های نوع مرتبه اول	
۳- قسمت سوم: زبان‌های برنامه‌نویسی تابعی (۴ هفته) (a) بررسی زبان‌های اعلانی/دستوری/تابعی/شیء‌گرا (b) Syntax در مقابل Semantics (در مقابل idioms در مقابل libraries در مقابل tools) (c) اصول ML (پیوندها، Shadowing، توابع) (d) Pairs, Tuples, Lists (e) مزایای عدم تغییر در مقابل تغییر (f) Record & Type Binding (g) کلاس مرتبه اول در مقابل سطح بالاتر و در مقابل توابع ناشناس (h) Function Closures و محدوده Lexical (i) Type inference & Orthogonality (j) انواع داده و ماژول‌ها (k) عبارات‌ها و محدوده‌های Let (l) موضوعات پیشرفته در برنامه‌نویسی کاربردی (کنترل، خطا، ادامه، ...)	
۴- قسمت چهارم: زبان‌های برنامه‌نویسی شیء‌گرا (۴ هفته) (a) مقدمه‌ای بر زبان‌های برنامه‌نویسی شیء‌گرا (b) کلاس‌ها و اشیاء (c) برنامه‌نویسی شیء‌گرا در مقابل تجزیه و تحلیل کاربردی، توسعه پذیری و پیاده‌سازی (d) انتزاع داده و Modularity (e) چهار مفهوم اساسی در زبان‌های شیء‌گرا (f) ساختار برنامه و الگوی طراحی (g) ++C: مطالعه‌ای مختصر در مورد اشیاء، وراثت، و کارایی زمان اجرا (h) JAVA: مطالعه‌ای مختصر در مورد معماری سیستم، امنیت، قابلیت انتقال و ایمنی	
۵- قسمت پنجم: همروندی (۲ هفته)	



<p>(a) مفاهیم پایه در همروندی (b) مدل actor (c) مقایسه همروندی: ML در مقابل جاوا (d) مروری سریع بر برنامه‌نویسی منطقی</p>	
	نرم افزارها و ابزارهای مورد نیاز
<p>هفت تکلیف تکلیف ۱: مفاهیم پایه تکلیف ۲: محاسبات لامبدا و منطق زبان‌های برنامه‌سازی تکلیف ۳: زبان‌های برنامه‌نویسی تابعی تکلیف ۴: سیستم نوع و استنتاج نوع تکلیف ۵: زبان‌های برنامه‌نویسی شی‌گرا تکلیف ۶: جاوا در مقابل C++ تکلیف ۷: همروندی</p>	تکالیف پیشنهادی
<p>سه پروژه برنامه‌نویسی برای پیاده‌سازی بخش‌هایی از یک زبان کوچک با استفاده از معانی آن: مرحله اول: جایگزینی نام (با استفاده از توابع) مرحله دوم: پیاده‌سازی یک زبان برنامه‌نویسی بسیار کوچک (با استفاده از سیستم‌های نوع) مرحله سوم: پیاده‌سازی subtyping</p>	پروژه‌های پیشنهادی
<p>تکالیف ۱۷,۵٪ (۷ تمرین) پروژه‌ها ۷,۵٪ (۳ فاز) کوئیز ۱۰٪ (۴ کوئیز) امتحان‌های میان ترم ۳۰٪ (۲ امتحان) امتحان پایان ترم ۳۵٪</p>	نمره‌دهی پیشنهادی
<p>[1] Programming Languages Course by Dan Grossman, University of Washington, Spring 2014, available on Coursera. [2] Concepts of programming languages by Robert W. Sebesta, Addison-Wesley Publishing Company, 11th edition, 2015. [3] Type Systems by Luca Cardelli, Microsoft Research, CRC Handbook of Computer Science and Engineering, 2nd Edition, Ch. 97, 2004.</p>	سایر مراجع
<p>مهران علیدوست‌نیا</p>	تنظیم کننده
<p>۱۱ مهر ۱۳۹۶</p>	تاریخ تنظیم