

درس تخصصی گرایش نرم‌افزار: طراحی زبان‌های برنامه‌سازی

نام درس	طراحی زبان‌های برنامه‌سازی		
نام درس به انگلیسی	Design of Programming Languages		
نوع واحد	تخصصی	مهندسی کامپیوتر	۳ واحد
مقطع	کارشناسی		
هم‌نیازها			
پیش‌نیازها	اصول طراحی کامپایلر		
مطالب پیش‌نیاز	آشنایی با حداقل یک زبان برنامه‌نویسی شیء‌گرا - مبانی کامپایلر		
کتاب(های) مرجع	[1] John Mitchell, <i>Concepts in Programming Languages.</i> , Cambridge University Press, 2004.		
اهداف درس	<p>هدف این درس آشنایی دانشجویان با روش‌های برنامه‌سازی، مفاهیم و قابلیت‌های زبان‌های برنامه‌نویسی و همچنین تاریخچه و چگونگی پیاده‌سازی آن‌هاست. یادگیری مفاهیم و قابلیت‌های برنامه‌نویسی به دانشجویان کمک می‌کند که ویژگی‌های یک زبان را از کنار هم فرادادن قابلیت‌های آن استنتاج کنند و در نتیجه زبان‌های جدید را راحت‌تر فراگیرند و از قابلیت آن‌ها بهتر و بهینه‌تر استفاده نمایند. همچنین هزینه هر قابلیت زبان را درک کنند و در انتخاب زبان برنامه‌نویسی برای هر کاربرد خاص، انتخاب بهتری از میان گزینه‌ها با توجه به روش برنامه‌سازی و نیازهای برنامه‌نویسی داشته باشند.</p>		
نتایج درس	<p>دانشجویانی که این درس را با موفقیت پشت سر بگذارند بینش مناسبی در موارد زیر خواهند داشت:</p> <ol style="list-style-type: none"> ۱- امکان‌سنجی پیاده‌سازی دستورات زبان‌ها توسط کامپایلرها ۲- زبان / رویکردهای برنامه‌نویسی مانند تابعی، رویه‌ای و شیء‌گرا. ۳- ویژگی و قابلیت‌های هر رویکرد زبان برنامه‌نویسی و چالش‌های پیاده‌سازی آن‌ها. 		
فهرست مباحث	<ol style="list-style-type: none"> ۱- انواع زبان‌های برنامه‌نویسی: زبان‌های اعلانی (declarative)، دستوری (imperative) ۲- تاریخچه زبان‌ها: Fortran, Cobol, Algol 60, Algol 68, Pascal, Modula, B, C, LISP. ۳- قدرت زبان‌های برنامه‌نویسی و نظریه محاسبه‌پذیری ۴- حسابان لامبدا و زبان LISP ۵- زبان ML ۶- بررسی گونه‌ها: بررسی گونه ایستا/ پویا، استنتاجی و استقرایی، کلاس گونه در Haskell در کنترل overloading توابع ۷- مدیریت حوزه: مفاهیم بلوک و ساختار پشته، رکورد فعالیت، مدیریت حوزه به‌صورت ایستا و پویا، پیاده‌سازی ساختار بلوک‌های تودرتو با استفاده از پیوند کنترل، پیاده‌سازی تابع‌های درجه اول (first-order) با استفاده از پیوند دسترسی و تابع‌های درجه دوم (higher-order) با استفاده از closure، بهینه‌سازی با استفاده از tail recursion، پیاده‌سازی thunk (call-by-need) ۸- ساختارهای کنترلی: پیاده‌سازی exception، آشنایی با exception در زبان ML، پیاده‌سازی continuation و کاربرد آن در کرنل‌نویسی سیستم‌عامل‌ها و کامپایلرها ۹- زبان‌های مدولار: مفاهیم اولیه (Interface, Specification, Implementation)، انتزاع داده و رویه، نوع داده انتزاعی، package، generic abstraction، template در زبان C++ ۱۰- خصیصه‌های زبان‌های شیء‌گرا: encapsulation، dynamic look-up، subtyping، inheritance ۱۱- آشنایی با زبان‌های شیء‌گرای Smalltalk, Simula, C++, Java و تفاوت آن‌ها در پیاده‌سازی 		



<p>خصوصه‌های شی‌گرایی، مشکلات و مزید کاربرد هر یک از آن‌ها مانند وراثت چندگانه در ++C، اجرای پویا در جاوا</p> <p>۱۲- زبان‌های همروند و توزیع شده: Actor، پیاده‌سازی همروندی ML با استفاده از مفهوم کانال، ریشه‌ها در Java</p>	
<p>برنامه‌نویسی در یکی از زبان‌های تابعی مانند Standard ML</p>	<p>نرم‌افزارهای مورد نیاز</p>
<p>۶ تکلیف دستی و ۳ تمرین کامپیوتری: در این درس دانشجویان چندین تمرین کامپیوتری با زبان‌های LISP، ML و Java انجام می‌دهند.</p>	<p>تکالیف پیشنهادی</p>
-	<p>پروژه‌های پیشنهادی</p>
<p>۲۵٪ تکالیف</p> <p>۷۵٪ آزمون‌ها</p>	<p>نمره‌دهی پیشنهادی</p>
<p>[1] D. Friedman and M. Wand, <i>Essentials of Programming Languages</i>. The MIT press, third edition, 2008.</p> <p>[2] M. Gabrielli and S. Martini, <i>Programming Languages: Principles and Paradigms</i>. Springer Publications Inc., 2010.</p>	<p>سایر مراجع</p>

