



# طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

## Programming Languages: Design and Implementation

مدرس: محمد ایزدی

پیش‌نیاز: اصول کامپایلرها

واحد: ۳

گرایش: نرم افزار

مقطع: کارشناسی

### کلیات

هدف از ارائه این درس آشنایی دانشجویان با اصول و مسایل مرتبط با طراحی زبان‌های برنامه‌سازی و همچنین روش‌ها و ساختارهای داده‌ای به کار رفته در پیاده‌سازی یا محقق کردن محیط برنامه نویسی و اجرای برنامه‌ها با زبان‌های برنامه‌سازی در سنت‌های موجود است. همچنین یکی از اهداف این درس آشنایی دانشجویان با سیر تحولات مفاهیم و روش‌های طراحی و پیاده‌سازی نسل‌های مختلف زبان‌های برنامه‌سازی از دیدگاه تاریخی و آشنایی با زبان‌های اصلی که موجب ایجاد و یا تحولات اساسی در این سنت‌های برنامه‌سازی شده‌اند می‌باشد. به طور کلی سنت‌های برنامه‌سازی یا روش‌های تعریف مفهوم برنامه که بر اساس آنها زبان‌های برنامه‌سازی مناسب طراحی می‌شوند را می‌توان به چهار دسته تقسیم کرد: سنت برنامه‌سازی امری-رویه‌ای (Imperative-Procedural)، سنت برنامه‌سازی شی‌گرا (Object Oriented)، سنت برنامه‌سازی منطقی یا مبتنی بر قاعده (Rule Based) و سنت برنامه‌سازی تابعی (Functional). با توجه به اهداف بالا و نظر به آن که دانشجویانی که این درس را اخذ می‌کنند قبلاً با برنامه‌سازی به روش‌های امری-رویه‌ای، شی‌گرا و مبتنی بر قاعده آشنایی نسبی پیدا کرده‌اند ارائه این درس به دو نیم ترم تقسیم می‌شود. در نیم ترم اول مروری بر مفاهیم اصلی در طراحی زبانها و برخی زبانهای مهم ارائه خواهند شد. منبع اصلی این بخش کتاب MacLennan خواهد بود. نیم ترم دوم به ارائه تجربی یک زبان برنامه‌سازی تابعی (زبان Scheme) و طراحی و پیاده‌سازی مرحله به مرحله آن اختصاص دارد. در نتیجه نه تنها به صورت نظری که در عمل نیز دانشجویان با پیاده‌سازی زبانهای برنامه‌سازی تابعی آشنا می‌شوند. منبع اصلی این بخش کتاب Friedman خواهد بود. ضمناً در ارائه این درس فرض بر آن است که دانشجویان با یکی از زبان‌های برنامه‌سازی C++ و Java به طور کامل و با زبان‌های C و Prolog آشنایی اجمالی دارند.

### ریز مواد

- مقدمه - معرفی شامل تاریخچه تکاملی زبانهای برنامه‌سازی، معرفی سنتهای برنامه‌سازی و مسایل اصلی در پیاده‌سازی زبانها
- معرفی تکاملی سیر تحول مفاهیم اصلی برنامه‌سازی در نسل‌های متکامل شده زبان‌ها به ترتیب شامل:
  - Fortran
  - Algol
  - Pascal
  - ADA
  - Common LISP
  - Smalltalk
  - Prolog
- مروری بر روش‌های پیاده‌سازی و کنترل روند اجرای برنامه‌ها. پیاده‌سازی ساختار بلوکی، کنترل فراخوانی و اجرای زیربرنامه‌ها و مدیریت حافظه.
- معرفی اجمالی برخی زبان‌های برنامه‌سازی امروزی‌تر مانند ML، Python و Haskell.

۵. مروری بر نیازمندی‌هایی از سایر زمینه‌ها که بر طراحی و پیاده‌سازی زبان‌ها تاثیر می‌گذارند مانند نیازمندی‌های برنامه‌سازی موازی یا هم‌روند، بلادرنگ بودن، تحت web بودن و مهندسی نرم‌افزار مولفه‌گرا (Component Based SE) یا سرویس‌گرا (Service-Oriented SE).
۶. طراحی و پیاده‌سازی تجربی و گام به گام یک زبان برنامه‌سازی تابعی (مانند Scheme) که شامل فصل‌های ۱ تا ۶ کتاب Friedman خواهد بود.

### آزمون، تمرین و گزارش پژوهشی

- آزمون: آزمونهای نیم ترم (۳۰٪ کل نمره) و پایان ترم (۴۰٪ کل نمره).
- تمرین: چند سری تمرین جهت آشنایی با برنامه‌های تابعی (۱۵٪ کل نمره).
- گزارش پژوهشی: ارایه‌ای از یکی از زبان‌های برنامه‌سازی یا زبان‌های توصیف نرم‌افزار که در کلاس مورد بحث قرار نمی‌گیرند (۱۵٪ کل نمره).

### مراجع اصلی

- [1] Daniel P. Friedman and Mitchell Wand, *Essentials of Programming Languages*, 3rd Edition, MIT Press, 2008.
- [2] Bruce J. MacLennan, *Principles of Programming Languages Design, Evaluation, and Implementation*, 3rd Edition, Oxford University Press, 1999.
- [3] Terrence Pratt, Marvin Zelkowitz, *Programming Languages: Design and Implementation*, 4<sup>th</sup> Edition, Prentice-Hall, 2000.