به نام خدا

طرح دوره‌ی معماری نرم‌افزار

پاییز ۱۴۰۰

فهرست

۱. مدرس

محمدرضا طیبی (www.tyyi.net)

۲. سوابق مدرس

- کارشناسی مهندسی کامپیوتر – دانشگاه بوعلی سینا
- فعال بخش خصوصی در صنعت فناوری اطلاعات
- مفتخر به کسب برخی از مقام‌ها و حضور در جشنواره‌ها و مسابقات ملی و جهانی

۳. برگزار کننده

۴. شرح

به **فرایند شکستن یک سامانه‌ی نرم‌افزاری به زیر سامانه‌ها و قطعات، تعیین سیاست‌های تعامل این بخش‌ها، و تعیین واسط‌های آن‌ها**، معماری نرم‌افزار گفته می‌شود. این دانش، بر اساس اصولی از مهندسی نرم‌افزار، سیستم‌های توزیع شده، پایگاه‌های داده و سیستم‌های اطلاعاتی، تئوری پیچیدگی، و مدل‌سازی بازدهی بنا شده است.

این دوره با رویکرد **جز به کل**، پس از مروری بر مبانی برنامه‌نویسی، به سمت مباحث پیشرفته‌تر در توسعه‌ی نرم‌افزار حرکت کرده، و در نهایت با بررسی معماری نرم‌افزارهای سازمانی، و مباحث پیشرفته در معماری نرم‌افزار به کار خود خاتمه می‌دهد.

۵. هدف

معماری نرم‌افزار، مهم‌ترین تأثیر را در **کیفیت و قیمت یک نرم‌افزار** دارد. هر ۱۰ سال، تعداد نرم‌افزارهای تولید شده ۱۰ برابر می‌شود و همچنین پیچیدگی هر نرم‌افزار به مرور زمان افزایش میابد. در نتیجه معماری نرم‌افزار هر روز از اهمیت ویژه‌تری برخوردار است. در مدل کلاسیک توسعه‌ی یک نرم‌افزار که از مراحل ۱- نیازسنجی و معماری، ۲- طراحی، ۳- پیاده‌سازی، ۴- تست، ۵- تحویل، و ۶- نگه‌داری و استفاده تشکیل شده است، تنها ۱۰ درصد از زمان به معماری اختصاص دارد اما تعیین کننده‌ی ۹۰ درصد از هزینه‌ها و ریسک‌هاست.

هدف از این دوره **تربیت برنامه‌نویس‌های آشنا به اصول معماری نرم‌افزار** برای کاهش هزینه‌ها و ریسک‌های سامانه‌های نرم‌افزاری است.

این دوره تلاش می‌کند تا مستقل از زبان برنامه‌نویسی عمل کند. در طول دوره مثال‌هایی از زبان‌های برنامه‌نویسی مختلف ارائه خواهد شد. مدرس می‌تواند به صورت حرفه‌ای به حل مشکل‌های مربوط به زبان‌های **سی‌شارپ** و **پی‌اچ‌پی**، در در سطح غیر حرفه‌ای می‌تواند به پرسش‌های زبان‌های جاوااسکریپت، گولنگ، و با کمک موتورهای جستجو و دریافت فرصت برای مطالعه، به حل مسائل سایر تکنولوژی‌ها بپردازد.

هدف از مرور سریع بر مبانی برنامه‌نویسی، آن است که قابلیت‌های ابزارها و شیوه‌ی کار ابزارهای مختلف را بهتر بشناسیم.

۶. پیش‌نیاز

دانشجویان بهتر است پیشتر به صورت خودآموز و یا آکادمیک، با برنامه‌نویسی آشنا باشند. چرا که تنها به مرور موارد بسنده خواهیم کرد. تمریناتی که احتیاج به پیاده‌سازی نرم‌افزاری دارند، نیاز به مهارت و سرعت عمل در برنامه‌نویسی دارند.

در صورتی که دانشجویان از قبل با برنامه‌نویسی آشنا نیستند، مسیر دشواری را طی خواهند کرد و باید تا پیش از پایان بخش‌های مربوط به مفاهیم برنامه‌نویسی، با استفاده از دنبال کردن کلیدواژه‌ها و منابع معرفی شده، مهارت و دانش خود را ارتقا دهند.

تحصیلات مهندسی کامپیوتر، مهندسی برق، مهندسی بیو انفورماتیک، فیزیک، و ریاضی، می‌تواند کمک‌کننده باشد.

۷. سیاست‌های دوره

۷.۱ تمرین

حل تمرین مهم‌ترین عامل دستیابی به اهداف آموزشی این دوره است. تمرین‌ها گاهی اوقات مسائل انتزاعی و گاهی اوقات پروژه‌های واقعی در اندازه‌های متفاوت هستند. دامنه‌ی مسئله‌ها گاهی بسیار جزئی و گاهی بسیار بزرگ است. هدف از تمرین‌ها، آمادگی ذهنی و دستیابی به دیدگاه مناسب در مورد هر بخش از آموزش است.

۷.۲ تمرکز در کلاس، و ارائه‌ی جزوه، و ضبط چندرسانه‌ای

طراحی این دوره به نحوی است که با مشارکت دانشجویان در بحث‌های آزاد و حل چالش‌ها به صورت زنده به برخی از اهداف آموزشی خود می‌رسد. در خصوص ضبط محتوای کلاس نسبت به **سیاست‌های برگزارکننده‌ی دوره** اقدام خواهد شد؛ هرچند فیلم‌ها نه می‌توانند تمامی آنچه در کلاس رخ می‌دهند را منتقل کنند، و حتی در آن شرایط نیز، نمی‌توانند محیط و تمرکز لازم برای به خاطر نشستن مفاهیم آموزشی را فراهم کنند. در نتیجه توصیه می‌کنیم که دانشجویان با تمرکز کامل، همراه داشتن آب یا قهوه یا نوشیدنی‌های قندی، خوراک خشک با بسته‌بندی مناسب و راحت در کلاس حاضر شوند. بهتر است که وعده‌های اصلی غذا، قرارهای ملاقات و سایر عوامل مضطرب‌کننده را در حاشیه‌ی امنی بیشتر از یک ساعت با زمان شروع و پایان کلاس قرار دهند.

اسلایدهای دوره‌ها در وب‌سایت مدرس به اشتراک گذاشته می‌شود و سعی می‌شود که کلیدواژه‌ها در اسلایدها آورده شود تا بتوان راحت‌تر مطالب کلاس را «مرور» کرد. اسلایدها، جایگزین تمرین‌ها، کتاب‌ها، و سایر منابع آموزشی معرفی شده نیستند.

۷.۳ غیبت

چنانچه دانشجویان به جهت بیماری، بدی آب و هوا، مراسم‌های شادی (فقط اقوام نزدیک) و سوگ، یا قراری که نباید از دست برود، جلسه‌ای از کلاس را از دست دادند، نیاز به ارائه‌ی دلیل به مدرس نیست. هرچند سیاست‌های برگزار کننده‌ی دوره به قوت خود باقی است.

۷.۴ تفاوت‌های جسمانی (معلولیت)

دانشجویان در صورتی که دارای تفاوتی در جسم خود هستند که نیاز به فراهم کردن شرایط ویژه‌ای دارد، با برگزار کننده و مدرس در ارتباط باشند تا راه‌کارهای لازم بررسی شوند.

۸ جامعه‌ی هدف

- توسعه‌دهنده‌های تازه‌کار که تمایل دارند ارشد (Senior) باشند؛
- توسعه‌دهنده‌های سنیور و سایر افرادی که **همچنان معتقدند روایت‌های مختلف از معماری نرم‌افزار، ارزش شنیدن دارد**؛
- افرادی که برنامه‌نویسی را نه‌تنها به عنوان شغل، بلکه ابزاری برای خلق ارزش و ابتکار می‌دانند؛
- افرادی که دوست دارند موارد جدیدی بیاموزند؛
- افرادی که دوست دارند مروری بر آنچه می‌دانند داشته باشند.

۹ طول دوره

طول دوره ۳۰ جلسه ۹۰ دقیقه‌ای خواهد بود. هر جلسه یک استراحت ۱۵ دقیقه‌ای در اواسط هر جلسه، بحث آزاد، و پرسش و پاسخ خواهد داشت.

۱۰ سرفصل‌های دوره

۱۰.۱ مروری سریع بر اصول برنامه‌نویسی

- قانون Moore در سخت‌افزار
- سلسله‌مراتب داده‌ها (Data Hierarchy)
- سطوح زبان‌های برنامه‌نویسی
- تکنولوژی Object و ایده‌ی استفاده‌ی مجدد از قطعات نرم‌افزار
  - Methods and Classes
  - Instantiation

- Reuse
- Messages and Method Calls
- Attributes and Instance Variables
- Encapsulation
- Inheritance

- محیط توسعه‌ی نرم‌افزار
  - فاز ۱: ویرایش
  - فاز ۲: پیش‌پردازش
  - فاز ۳: Compiler
  - فاز ۴: Linker
  - فاز ۵: Loader
  - فاز ۶: Instruction Execution
- مفاهیم اولیه
  - چاپ خروجی و دریافت ورودی
  - عملگرهای محاسباتی (Arithmetic) و اولویت عملیات
  - عملگرهای برابری و مقایسه
  - عملگرهای منطقی
  - عملگرهای بیتی (Bitwise)
- برنامه‌نویسی ساختارمند
  - الگوریتم
  - ساختارهای کنترل و goto-less programming
    - Sequence Structure
    - Selection Structure
      - *if*
      - *if...else*
      - *:?*

        - nested *if...else*
        - *switch...case*
    - Repetition Structure
        - w*hile*
        - *for*
        - *do...while*
        - *break*
        - *continue*
- تابع‌ها
    - Function Prototypes
    - Function Call Stack and Stack Frames
    - Headers
    - Passing Arguments By Value and By Reference
    - قوانین اسکوپ
    - Recursion vs. Iteration
    - Random Number Generation and Scaling
    - لیست آرگومان‌ها
- آرایه‌ها
    - پاس دادن آرایه به تابع
    - آرایه‌های چند بعدی
    - آرایه‌های با اندازه‌ی متغیر
    - جستجو در آرایه‌ها
        - جستجوی خطی
        - جستجوی دودویی
    - مرتب‌سازی آرایه‌ها
        - مرتب‌سازی حبابی (Bubble Sort)

        - مرتب‌سازی انتخابی (Selection Sort)
        - Insertion Sort
        - مرتب‌سازی سریع (Quick Sort)
        - مرتب‌سازی ادغامی (Merge Sort)
    - اشاره‌گرها
    - کاراکترها و رشته‌ها
    - فرمت ورودی و خروجی
    - Structها، Unionها، و Enumeration
    - کار با پرونده‌ها (File)
        - پرونده‌ها
        - Streamها
        - پرونده‌های دسترسی متوالی (Sequential-Access Files)
        - پرونده‌های دسترسی تصادفی (Random-Access Files)
    - داده‌ساختارها
        - ساختارهای خودمرجع (Self-Referential)
        - اختصاص پویای حافظه (Dynamic Memory Allocation)
        - لیست‌های پیوندی
        - استک‌ها
        - صف‌ها
        - درخت‌ها
    - پیش‌پردازش‌گر
        - فراخوانی کتاب‌خانه‌ها
        - ماکروها
        - کامپایل مشروط
        - Assertionها
        - کامپایل برنامه‌های Multi-Source-File

    - برای تفریح: ساختن یک زبان برنامه‌نویسی برای آدم فضایی‌ها!
- استفاده از ترمینال و خط فرمان
    - اجرای برنامه‌ها
    - دریافت ورودی از فایل
    - ارسال خروجی به فایل
    - پاس دادن آرگومان‌ها
- چند نخی و سرویس‌ها
    - Signal Handling
- شی گرایی
    - Overloading
    - Overwriting
    - Inline Functions
    - Lambda Functions
    - Default Arguments
    - Unary Scope Resolution Operator
    - کلاس
        - Polymorphism
        - Data Members
            - Getter and Setter Functions
            - *static / dynamic*
        - Member Functions
            - Constructor Function
            - *const*
            - *this*
            - *virtual*
        - Access Specifiers

        - *public*
        - *protected*
        - *private*
    - OOAD
- مدیریت خطا
    - throw (Rethrowing Exception)
    - throw (Stack Unwinding)
    - *try...catch*
- ...

۱۰.۲. آزمون نرم‌افزار

- Unit Test
- Test Driven Design

۱۰.۳. مروری بر کدِ تمیز

- اصطلاح‌ها
    - اثر جانبی (side-effect)
    - Good Practice
    - Best Practice
    - Portability
    - Performance
- قانون KISS
- قانون DRY
- قوانین SOLID
    - قانون Single Repository
    - قانون Open to extension / Closed to modification
    - قانون Liskov Substitution

    - قانون Interface Segregation
    - قانون Dependency Inversion
- قانون Composition Over Inheritance
- قانون Single Responsibility
- قانون You Aren’t Going to Need It
- قانون Easy to read, easy to maintain
- مستندسازی کد
- پرداخت (Refactor)
- اسم‌های معنی‌دار
- توابع
- توضیحات (Comments)
- فرمتینگ (Formatting)
- اشیاء و داده‌ساختارها
- مدیریت خطا
- تست‌ها

۱۰.۴. فرابرنامه‌نویسی (Meta Programming) و الگوهای طراحی (Design Pattern)

۱۰.۵. مهندسی نرم‌افزارهای خیلی بزرگ (VLSE)

- معماری نرم‌افزار چیست؟
- نرم‌افزارهای کاربردی (Application)
- نرم‌افزارهای سازمانی (Enterprise)

۱۰.۶. مفاهیم پایه

- تقلیل‌دهی (Abstraction)
- واسط‌ها و واسط‌های برنامه‌های کابردی (Interfaces and APIs)

- تفکیک دغدغه‌ها (SoC)
- مهندسی نیازمندی‌ها
- فرا برنامه‌نویسی (Metaprogramming)
- شفافیت، جعبه سفید، و لایه‌ی تقلیل‌یافتگی
- یکپارچه‌سازی (Integration)
- هم‌کنش‌پذیری (Interoperability)
- هم‌بستگی (seamlessness)
- معماری مونولیتیک / ماژولار (monolithic/modular)
- نقش معمار نرم‌افزار
- قطعه (component)
- پیونددهنده (connector)
- ترکیب‌بندی (composition)
- نمایه (view)
- جزیره‌بندی (federation)
- وابستگی‌ها (dependencies)
- رویداد محوری (event-driven)
- UML
- میان‌افزار (middleware)

۱۰.۷ معماری عملگرا (Functional Architecture)

- Model-View-Controller
- لفاف‌پیچی (wrapper)
- پراکسی (proxy)
- کشینگ (caching)

۱۰.۸ معماری جنبه‌گرا (Aspect-Oriented Architecture)

- امنیت

- اتکاپذیری (reliability)
- مقیاس‌پذیری (scalability)
- تحول‌پذیری (evolvability)
- قابلیت بقا (survivability)
- قابلیت استفاده (usability)
- قابلیت نگه‌داری (maintainability)
- بازدهی (performance)
- دسترس‌پذیری (availability)
- سازگاری با گذشته (backward compatibility)
- ضمانت کیفیت خدمات (quality of service guarantee)
- توافق‌های سطح خدمت (service-level agreements)
- قوانین کسب‌وکار (business rules)
- سیاست‌ها (policies)

۱۰.۹. رایانش توزیع‌شده

- RPC
- client-server
- همتا به همتا (peer-to-peer)
- رایانش مشبک (grid-computing)
- رایانش ابری
- معماری اشتراکِ هیچ (Shared Nothing Architecture)
- معماری خدمت‌گرا (service-oriented architecture)
- جریان‌کاری
- جریان‌کاری مشبک
- WSDL
- SOAP

- UDDI
- REST
- افزونه‌ها

۱۰.۱۰. معماری‌های Domain Specific

- Domain Driven Design

۱۰.۱۱. معماری تمیز

- قطعات (Components)
  - انسجام (Cohesion)
  - استقلال (Independence)
- درس‌هایی از معماری تمیز

۱۱. منابع دوره

- **BOOK: HOW TO PROGRAM C, SEVENTH EDITION;** *PAUL DEITEL, HARVEY DEITEL*
- **BOOK: CLEAN CODE, A HANDBOOK OF AGILE SOFTWARE CRAFTSMANSHIP;** *ROBERT C. MARTIN*

  BOOK: CLEAN ARCHITECTURE, A CRAFTSMAN'S GUIDE TO SOFTWARE STRUCTURE AND DESIGN; *ROBERT C. MARTIN*
- **COURSE: DEPARTMENT OF COMPUTER SCIENCE AND COMPUTER ENGINEERING, COLLEGE OF ENGINEERING, UNIVERSITY OF ARKANSAS;** CSCE 5013 Software Architecture; *Craig Thompson*

۱۲. شیوه‌ی برگزاری

- برگزاری این دوره به صورت حضوری خواهد بود.
- برخی از جلسات که تمرین با کامپیوتر دارند، بهتر است که دانشجویان لپ‌تاپ شخصی‌سازی شده به همراه داشته باشند. در غیر این صورت می‌توانند از امکانات برگزار کننده‌ی دوره استفاده کنند.

۱۳. تقویم دوره