CS 671: Programming Language Concepts and Features

Catalog description

Explores the main features of modern, high-level, general-purpose programming languages from the user (programmer) standpoint. Students learn how specific features of programming languages can be used effectively in solving programming problems. The course is also an opportunity to use paradigms that expand on simple imperative programming, such as object-oriented, functional and concurrent programming. Prereq: CS520, some knowledge of Java.

Overview

Everything that is computable can be computed using very few basic building blocks, like assignments, conditionals, loops and recursion. In practice, however, modern, high-level programming languages offers a profusion of features that help implement, maintain and extend large, complex pieces of software: support for object-oriented, functional, concurrent and distributed programming, modularity, code reuse, types, exception handling, pattern matching, etc. This course explores the main features of modern programming languages from a user's standpoint, by illustrating their use through practical examples and programming assignments.

Most of these features can be found in one form or another in many modern languages, like Java, Kotlin, Python, Ruby, Clojure, Rust, Javascript, C#, F#, etc. Although some of these languages are discussed when relevant, all implementations are done in Scala, a modern cousin of Java that runs on the Java Virtual Machine and that integrates many of the most interesting features of modern programming languages. Scala provides us with a platform on which to experiment with many different features within a single language. The course is programming intensive and a substantial part of the final grade is based on programming assignments.

Attributes

- This course is *required* for CS majors. The minimum passing grade is D^-.

Evaluation

Five programming assignments (50%), 7 quizzes (30%) and one exam (20%).

Minimum score for each grade: A: 90, A^-: 87, B^+: 83, B: 80, B^-: 77, C^+: 73, C: 70, C^-: 67, D^+: 63, D: 60, D^-: 57.

ABET Outcomes

- **Outcome 2:** students write programs using modern programming language features; these programs are thoroughly evaluated for correctness and performance.

ABET Curriculum

- **Curriculum 1:** programming techniques; testing and debugging skills; standard tools (version control, unit testing and IDE).
- **Curriculum 4:** a) recursive and concurrent algorithms; c) programming language features.

- **Curriculum 5:** in-depth coverage of the Scala programming language.
- **Curriculum 6:** e) introduction to multi-threaded programming.

Topics

- **Basic syntax and semantics:**
  - values, expressions, primitive types, identifiers, variables, scope
  - lists, arrays, tuples, options
  - conditionals, loops, exceptions, iterators, pattern-matching
  - objects, methods, functions, parameters, types
  - infix operators, string interpolation
- **Intermediate syntax and semantics:**
  - modules, packages, imports, collections
  - preconditions, postconditions, assertions
  - parametrized methods and types, type inference
  - variable-length arguments, default values, pass-by-name arguments
- **Advanced syntax and semantics:**
  - implicit arguments, implicit conversions
  - user-defined pattern-matching (case classes, extractors)
  - type bounds, wildcards, covariance and contravariance
  - domain specific languages
  - self-types and collections design
- **Object-oriented programming:**
  - classes, objects, singletons
  - interfaces, traits, inheritance, overriding, abstract members
  - subtyping, Liskov substitution principle, polymorphism, dynamic binding, “mixins”
  - encapsulation, equality
- **Functional programming:**
  - immutability, recursion, tail recursion, memoization, dynamic programming
  - first-class functions, function literals, higher-order functions, Currying, closures
  - streams, lazy evaluation
  - monads
- **Concurrent programming:**
  - threads, synchronization
  - thread pools, futures, promises
  - functional concurrent programming

Textbook (reference)

- Martin Odersky, Lex Spoon and Bill Venners. *Programming in Scala*, 4th edition, Artima Press, 2019. ISBN: 978-0-9815316-1-8.