CS464 Programming Languages

Course number and name: CS464: Programming Languages
Credits and contact hours: 3.000 Credit Hours, 3.000 Lecture hours
Instructor or course coordinator: Fatemeh Abdollahzadeh
Other supplemental materials: ACM SIGPLAN NOTICES

Course description: The course provides a comprehensive coverage of the fundamental concepts of programming languages by discussing the design issues of the various languages constructs, examining the design choices for these constructs in some of the most common languages, and critically comparing design alternatives. It discusses the formal methods of describing the syntax and semantics of programming languages. It discusses the main components of the computer and the basic principles of its operation.

Course Prerequisites: CS253

Prerequisites by Topics:
Familiarity and complete understanding of at least one high level language like C, C++, JAVA, or Pascal.
Concept and techniques for structuring and manipulating data.

Specific goals for the course:

a. The student will be able to explain the differences between different programming languages constructs.

   Course outcome 1: The student will understand and learn how to write scanner and parser for a context free grammar.
   Course outcome 2: The student will learn Lisp Language by writing some program.
   Course outcome 3: The student will learn the primary formal method for describing the syntax of programming language – BNF.
   Course outcome 4: The student will learn how programming languages are build out of syntax and semantic description of the language.

b. The following outcomes are addressed by the course.

   • Outcome (h): Recognition of the need for and an ability to engage in continuing professional development.
   • Outcome (f): An ability to communicate effectively.
• Outcome (i): An ability to use current techniques, skills, and tools necessary for computing practice.
• Outcome (j): An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices.
• Outcome (k): An ability to apply design and development principles in the construction of software systems of varying complexity

**Major topics Covered in the course:**
1. The role of programming of programming languages (1.5 hours)
2. Language description: syntactic structures (3 hours)
3. Statements: structured programming (3 hours)
4. Types: Data representation (4.5 hours)
5. Procedure activations (5 hours)
6. Grouping of data and operations (5 hours)
7. Object oriented programming (5 hours)
8. Element of functional programming, (3 hours)
9. Functional programming in a typed language (3 hours)
10. Functional programming with lists (3 hours)
11. An introduction to concurrent programming (2 hours)

**Laboratory Projects:**
1. Few programs to understand the structures of different languages and parse (scanner and syntax analysis) a context free grammar.
2. Project. Every student is required to submit one written report (not including exams, tests, quizzes, or commented programs) and to make oral presentation.

**Expected Performance Criteria**
Students will demonstrate their mastery of structure of programming languages in few programming and a project. The student’s understanding of structure of programming languages will be demonstrated in written answers to questions on midterms and the final.

**Grading Policy:** Two Midterm exams (%50), final exam (%30), and assignments (%20)

<table>
<thead>
<tr>
<th>Total points, final grade:</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
</tr>
<tr>
<td>----</td>
</tr>
<tr>
<td>95-100</td>
</tr>
</tbody>
</table>