CSC 340 Syllabus

First term 2019-2020

Course title: Programming Languages & Compilation Credit hours: 3
Prerequisites: CSC 212 and CSC 339
Instructor: Dr. Khalil El Hindi
Email: **khindi@ksu.edu.sa**
Room: 2122

Objectives

The objective of this course is to explore different types of programming languages and their features, and study translation/compilation techniques used in translating the high-level languages to a machine language. A basic compiler for a small programming language will be implemented during the semester.

Course Outcomes:

1. *The students will learn about the syntactic and semantic elements of programming languages.*
2. *The students will learn about grammars and their use in describing languages.*
3. *The students will learn the basic algorithms used by compilers to translate high-level programming languages into machine languages.*
4. *The students will learn how to implement a compiler through a programming project.*
5. *The students will learn to use compiler construction tools.*

Course Content:

1. Overview of Compilers and Programming languages
   A brief History of Programming Languages
   Factors that Influence Programming Language Designs
   Categories of Programming Languages

2. Lexical Analysis:
   Regular Expressions, Finite Automata, and Table Implementation

4. Parsing:
   Top-Down Parsing I
   LL(1) Parsing

Bottom-Up Parsing

LR(0), LR(1)

5. Semantic Analysis

Scoping

Type checking

6. Runtime Environments: Stack Machine

7. Code Generation

8. Local and global Optimization

Textbook & References:

1. *Engineering a Compiler* Second Edition, Elsevier
2. Concepts of Programming Languages by Robert Sebesta, Addison-Wesley.
3. Compilers: Principles, Techniques and Tools (2nd Ed.), by A. Aho, M. Lam, R. Sethi, J. Ullman, Addison Wesley.
4. Modern Compiler Implementation in Java, (2nd Ed.) by Andrew W. Appel and J. Palsber, Cambridge University Press
5. Compiler Construction: Principles and Practice, by Kenneth C. Louden, PWS Publishing

Expected Performance Criteria:

The students are expected to pass written examinations on class material, and complete a small project (arranged as a group of assignments) on implementing basic compiler for a small programming language.

Evaluation:

20% Midterm 1

25% Midterm 2

15% Project and Assignments

40% Final