Approval: 4th Senate Meeting

Course Name : Compiler Technology
Course Number : CS-402
Credits : 0-0-3-2
Prerequisites : None
Intended for : B.Tech.
Distribution : Compulsory for CSE; CS elective for EE and ME

Preamble: This is an elective course that builds on and gives a hands-on application of many previous courses like FLAT, Algorithms and Data Structures, Paradigms of Programming etc.
Course Outline: This course aims to provide the students with a thorough understanding of compilation technology. Assignments/Mini-project will provide the students with a practical knowledge of building compiler components using the open source LLVM framework. Towards the end, an overview of compiling functional languages will provide glimpses and illustrations of advanced programming language and compilation technology.
Pre-requisites:

1. Algorithms and Data Structures
2. Formal Languages and Automata Theory
3. Paradigms of Programming

Modules:
Unit 1: Introduction to compilers, Lexical Analysis, Syntax Analysis, Parsing – Top down, Bottom up and advanced, Syntax directed translation, Intermediate Code Generation – type checking and control flow.
Unit 2: Run time environments – Stack allocation, Heap management, Garbage Collection, Code generation and Optimization – machine independent, machine dependent, parallelism, data flow analysis and locality.
Unit 3: LLVM - Introduction, Illustration with examples, Mini-project.
Unit 4: Compiling Functional Languages – Review of Lambda Calculus, Translating functional programs into lambda calculus, Program representation and Graph reduction of lambda expressions, Supercombinators and lambda lifting, Advanced Graph Reduction – the G-Machine and Optimizations.
Unit 5 (Optional – time permitting): Compiling with Continuations – Review of Continuations and CPS, Conversion to CPS, Optimization of CPS, Closure Conversion and Machine code generation.

Textbooks:

1. A. Appel, Modern Compiler Implementation in C (Java, ML), Cambridge Univ. Press.
2. Simon Peyton-Jones, Implementation of Functional Languages, Prenctice-Hall.
3. A. Appel, Compiling with Continuations, Cambridge University Press.

References:

1. Aho, Lam, Sethi and Ullman, Compilers – Principles, Techniques, Tools, 2nd ed. Pearson/Addison-Wesley.