Compiler Theory Course Specifications

Faculty: Faculty of Computer and Informatics

Department: Computer Science

Program(s) on which the course is given : B. Sc. in Computer and Information Sciences

Major or Minor element of program : Computer Science

Department offering the program : Computer Science

Department offering the course : Computer Science

Academic year / Level : 4th Year/B.Sc.

Date of specification approval : 10/10/2009

A. Basic Information

Title: Compiler Theory **Code:** CSW 456

Lecture: 3 hrs/week **Tutorial:** 2 hr/week **Practical:** ---

Total: 5 hrs/week

B. Professional Information

1. Overall Aims of Course:

This course introduces student to the compiler design, as it is one of most important

system programs. The main motive to make students understand what happens behind the scene when they write application programs till it runs. The course comes as follows up on an operating system course and assumes knowledge of assembler design (prerequisite) to concentrate on specific issues of compiler construction.

2. Intended Learning Outcomes of Course (ILOs):

a. Knowledge and Understanding:

a1- Understand the main phases of compiler construction and how to select the appropriate approach for designing each phase.

a2- State regular expressions and draw their state diagrams NFA and DFA.

a3- Illustrate language syntax into context free grammar productions, and write attribute grammar to perform semantic actions on a CFG.

a4- Understand the overall structure of a compiler.

a5- Know about the grammar rules and syntax of languages.

b. Intellectual Skills:

b1- Formulate the whole process of compiler design with all its phases.

b2- Criticize languages according to their compilers and select the suitable language for each application.

b3- Identify attributes, components, relationships, patterns, main ideas, and errors for a program.

b4- Analyze ambiguities in language syntax and how to resolve them.

c. Professional and Practical Skills:

c1- Identify the best environment (compiling and linking options) for their programs to run most efficiently.

c2- Work effectively as an individual and as a member of a team.

c3- Perform independent information acquisition and management, using the scientific literature.

d. General and Transferable Skills:

d1- Presenting the whole process of designing and implementing a language; an experience that may help in applications.

d2- Learn how to work in groups and manage a project.

d3- Manage tasks and resources.

d4- Search for information and adopt life-long self-learning.

e. Attitude

e1- A knowledge and respect of ethics and ethical standards in relation to a major area of study.

e2- Demonstrate an ethical behavior toward software copyrights

e3- Explain the nature of privacy and how it is protected by the Data Protection.

3. Contents:

Topic	No. of hours	Lecture	Tutorial/ Practical
Over of system program in general, and the task of a compiler and its main phases.	5	3	2
The Lexical analyzer, main task, regular expressions, Finite State Machines.	5	3	2
The syntax analyzer, Grammars, Languages, and Pushdown Machines.	10	6	4
Top down parsing, Relations and Closure, Recursive Descent parser, LL(1) Grammars.	15	9	6
Bottom Up Parsing, Shift Reduce Parsing, LR Parsing With Tables, arrays.	15	9	6
Code Generation, Converting Atoms to Instructions, Single pass vs. Multiple passes, Register Allocation.	10	6	4
Optimization, Global Optimization Techniques, Local Optimization.	10	6	4