

SEMESTER	2
Paper Number	MCMS 4203
Paper Title	Compiler Design
No. of credits	6
Theory / Composite	Theory
No. of periods assigned	Th: 5      Tut: 1
Name of faculty member(s)	
Course description / objectives	<p>On completion of this course, the students will be able to:</p> <ol style="list-style-type: none"> <li>1. specify and analyse the lexical, syntactic and semantic structures of advanced language features</li> <li>2. separate the lexical, syntactic and semantic analysis into meaningful phases for a compiler to undertake language translation</li> <li>3. turn fully processed source code for a novel language into machine code for a novel computer</li> <li>4. describe techniques for intermediate code and machine code optimisation</li> <li>5. design the structures and support required for compiling advanced language features.</li> <li>6. understand the target machine's run time environment, its instruction set for code generation and techniques used for code optimisation</li> </ol>
Syllabus	<p>Review: Concepts and Types of Translators: Assembler, Cross-Assembler, Pre-Processor, Interpreter, Simulator, Loader; Linker. Grammars, Languages – types of grammars and their recognizers, Basic concepts of translators: boot strapping, cross compiler, Different phases of compilation.</p> <p>Lexical analyzer: Concepts, Tokens, Schemas, Design using FSM.</p> <p>Syntax Analysis: Top down and Bottom-up parser; Operator precedence; Recursive descent; LL (1); LR (1); LALR (1).</p> <p>Intermediate code generation: Three Address Code, Representation of three address code – Quadruples, Triples and Indirect Triples.</p> <p>Syntax directed translation: Attributes, Semantic Actions, Translation schemes.</p> <p>Code Optimization: Basic blocks, loop optimization, flow graph, DAG representations of basic blocks.</p> <p>Code generation: Object Programs, Problems in Code Generation.</p> <p>Error handling: detection, reporting, recovery and repair</p> <p>Symbol tables: Organization and management techniques.</p> <p>Runtime storage management: static allocation; dynamic allocation, activation records; heap allocation, recursive procedures</p>
Reading/Reference Lists	<ol style="list-style-type: none"> <li>1. Alfred V. Aho and Jeffrey D. Ullman, Principles of Compiler Design, Narossa Publication</li> <li>2. Aho, Sethi and Ullman, Compilers – Principles, Techniques and Tools, Narossa Publication</li> <li>3. Peter Linz, Formal Language and Automata Theory, Narossa Publication</li> <li>4. Systems Programming and Operating System, D. M. Dhamdhere, Tata McGraw Hills</li> <li>5. Systems Programming, John J Donovan, Tata McGraw Hills</li> </ol>
Evaluation	<p>Total – 100</p> <p>CIA – 20 Semester Examination – 80</p>