|  |  |
| --- | --- |
| **Course – 66 Title: Compiler Design and Automata Theory Sessional** |  |
| **Course No.: CIT-412 Credit : 1.5 Contact Hours: 2** | **Total Marks: 100** |

**11.1 Rationale**

Computer Engineers should be competent in compiler design and automata theory. They must learn the fundamental concepts of compiler design and automata theory and also various phases in the design of a compiler, how to generate a machine code from a C program statement.

**11.2 Objectives:**

* To design a Lex compiler
* To construct a DFA from the NFA
* To design an NFA for the corresponding regular expressions

|  |  |  |  |
| --- | --- | --- | --- |
| **11.3 Learning Outcomes** | **11.4 Course Content** | **11.5 Teaching  Learning Strategy** | **11.6 Assessment Strategy** |
| * Apply symbol table using c or java language | Symbol Table | * Demonstration * Exercise | * Assignment * Observation |
| * Apply & analyze lexical analyzer using Lex | Lexical analyzer using Lex | * Demonstration * Exercise | * Assignment * Observation |
| * Design DFA from the given regular expression | Construct DFA from regular expressions | * Demonstration * Exercise | * Assignment * Observation |
| * Design NFA from the given regular expression | Construct NFA from regular expressions | * Demonstration * Exercise | * Assignment * Observation |
| * Implement a C/Java program which eliminates the whitespace from the given source of code | A C/Java program which eliminates the whitespace from the given source of code | * Demonstration * Exercise | * Assignment * Observation |
| * Apply & analyze the design of a parser which produces parse tree for the tokens produced by the lexical analyzer | Design of a parser which produces parse tree for the tokens produced by the lexical analyzer | * Demonstration * Exercise | * Assignment * Observation |

**RECOMMENDED BOOKS AND PERIODICALS**

**Text Books**:

1. Alfred V.Aho, Ravi Sethi, Jeffrey D.Ullman : Compilers Principles, Techniques and tools. Third edition.
2. Hopcroft and Ullman : Introduction to Automata Theory, Languages and Computation

**References:**

1. Compiler Design Theory - by Lewis and Stern