RASHTRASANT TUKADOJI MAHARAJ NAGPUR UNIVERSITY, NAGPUR

FOUR YEAR BACHELOR OF TECHNOLOGY (B. Tech..) DEGREE COURSE

SEMESTER: VI (C.B.C.S.)

BRANCH: COMPUTER SCIENCE AND ENGINEERING

Examination Scheme and Syllabus

Sixth Semester:-

S. N.	Subject	Teaching Scheme			Evaluation Scheme			Credits	Category
		L	T	P	CA	UE	Total		
1	Compiler Design	4	-	-	30	70	100	4	PCC-CS
2	Compiler Design -Lab	-	-	2	25	25	50	1	PCC-CS
3	Elective-II	3	-	-	30	70	100	3	PEC-CS
4	Elective-III	3	-	-	30	70	100	3	PEC-CS
5	Open Elective-I	3	-	-	30	70	100	3	OEC
6	Professional Skills Lab II	-	-	2	25	25	50	1	PCC-CS
7	Hardware Lab	-	-	2	25	25	50	1	ESC
8	Mini Project	-	-	6	50	50	100	3	PROJ-CS
9	Economics of IT Industry	2	-	-	15	35	50	2	HSMC
10	Intellectual Property Rights (AuditCourse)	2	-	-	50	-	-	Audit	PCC
	Total	**17**	**-**	**12**			**700**	**21**	

Elective-II: - 1. Machine Learning 2. Internet of Things 3. Cluster and Cloud Computing

Elective-III: - 1. Data Science 2. Distributed Operating Systems 3. Human Computer Interaction

Open Elective 1:- 1. Linux Fundamentals 2. Android Application Development 3. Blockchain Technologies

RASHTRASANT TUKADOJI MAHARAJ NAGPUR UNIVERSITY, NAGPUR
FOUR YEAR BACHELOR OF TECHNOLOGY (B. Tech..) DEGREE COURSE
SEMESTER: VI (C.B.C.S.)
BRANCH: COMPUTER SCIENCE AND ENGINEERING

Subject: **Compiler Design** Subject Code: **BTECH_CSE-601T**

Load	Credits	College Assessment Marks	University Evaluation	Total Marks
48 Hrs	**4**	**30**	**70**	**100**

Aim: To understand the principles and concepts of Compiler Design

Prerequisite(s): Student should have basic knowledge of computers and mathematics.

Course Objectives:

1	Understand the phases of the Compiler and utilities of Automata.
2	Give the implementation details of Top-Down and Bottom-up Parsers and its types.
3	Describe the importance of the Semantic Phase and Symbol Table in Compiler.
4	Give the descriptions for the Synthesis Model of the Compiler w.r.t Analysis Model.
5	Understand the Architecture of the Computer and few advanced topics for a Compiler.

Course Outcomes:

At the end of this course students will be able to:

CO1	Define the Compiler along with phases and basic programs in LEX.
CO2	Develop programs for various kinds of the Parsers.
CO3	Write simple programs related to Type Checking, Parameter Passing and Overloading.
CO4	Implement the concepts of Code Optimizations and Code Generations.
CO5	Provide the Case Studies of Object-Oriented Compilers.

SYLLABUS:

UNIT-I:
Introduction: Phases of compilation and overview. Lexical Analysis (scanner): Regular languages, finite automata, regular expressions, relating regular expressions and finite automata, scanner generator (lex, flex).

UNIT-II:
Syntax Analysis (Parser): Context-free languages and grammars, push-down automata, LL(1) grammars and top-down parsing, operator grammars, LR(O), SLR(1), LR(1), LALR(1) grammars and bottom-up parsing, ambiguity and LR parsing, LALR(1) parser generator (yacc, bison)

UNIT-III:
Semantic Analysis: Attribute grammars, syntax directed definition, evaluation and flow of attribute in a syntax tree. Symbol Table: Basic structure, symbol attributes and management. Runtime environment: Procedure activation, parameter passing, value return, memory allocation,

UNIT-IV:
Intermediate Code Generation: Translation of different language features, different types of intermediate forms. Code Improvement (optimization): control-flow, data-flow dependence etc.; local optimization, global optimization, loop optimization, peep-hole optimization etc.

UNIT-V: Architecture dependent code improvement: instruction scheduling (for pipeline), loop optimization (for cache memory) etc. Register allocation and target code generation. Advanced topics: Type systems, data abstraction, compilation of Object Oriented features and non-imperative programming languages.

TEXT BOOKS:
1. Compilers: Principles, Techniques and Tools, V. Aho, R. Sethi and J. Ullman.
2. Lex&Yacc, Levine R. John, Tony Mason and Doug Brown

REFERENCES:
1. The Design and Evolution of C++, Bjarne Stroustrup.

RASHTRASANT TUKADOJI MAHARAJ NAGPUR UNIVERSITY, NAGPUR
FOUR YEAR BACHELOR OF TECHNOLOGY (B. Tech..) DEGREE COURSE
SEMESTER: VI (C.B.C.S.)
BRANCH: COMPUTER SCIENCE AND ENGINEERING

Subject: **Compiler Design Lab** Subject Code: **BTECH_CSE-601P**

Load	Credits	College Assessment Marks	University Evaluation	Total Marks
2 Hrs/Week	**1**	**25**	**25**	**50**

Course Objectives:

1	To learn usage of tools LEX, YAAC
2	To develop a code generator
3	To implement different code optimization schemes

Course Outcomes:

At the end of this course students will be able to:

CO1	Generate scanner and parser from formal specification.
CO2	Generate top down and bottom up parsing tables using Predictive parsing, SLR and LR Parsing techniques.
CO3	Apply the knowledge of YACC to syntax directed translations for generating intermediate code – 3 address code.
CO4	Build a code generator using different intermediate codes and optimize the target code.
CO5	Generate scanner and parser from formal specification.

Expected experiments to be performed (Not limited to):

1. Sample programs using LEX.
2. Scanner Generation using LEX.
3. Elimination of Left Recursion in a grammar.
4. Left Factoring a grammar.
5. Top down parsers.
6. Bottom up parsers.
7. Parser Generation using YACC.
8. Intermediate Code Generation.
9. Target Code Generation.
10. Code optimization