Programming Language and Compilers
Session: February'2022 - July'2022

1. **Course Number and Name:**

   CSE 4021, `Programming Language and Compilers`

2. **Credits and Course Format:**

   Grading Pattern = 1

   Credits = 4

   3 Classes/week, 1hr/Class,

   1 Problem Solving Session/Week, 2hr/Problem Solving Session

3. **Target Students:**

   Programme: B.Tech. (5th Semester)

   Branch: CSE, CS&IT

4. **Instructor's Names:**

   (1) **Dr. Niranjan panda**, Assistant Professor (CSE), ITER
   `niranjanpanda@soa.ac.in`

   (2) Dr. Bhupendra Kumar Gupta, Assistant Professor (CSE), ITER
   `bhupendragupta@soa.ac.in`

   (3) Dr. Deepak Patel, Assistant Professor (CSIT), ITER
   `deepakpatel@soa.ac.in`

   (4) Dr. Pravas Ranjan Bal, Assistant Professor (CSE), ITER
   `pravasbal@soa.ac.in`

   (5) Mr. Soumen Nayak, Assistant Professor (CSE), ITER
   `soumennayak@soa.ac.in`

   (6) Dr. Jitesh Pradhan, Assistant Professor (CSE), ITER
   `jiteshpradhan@soa.ac.in`

   (7) Ms. Trushna Parida, Assistant Professor (CSE), ITER
   `trushnaparida@soa.ac.in`

   (8) Mr. Sumit Kumar, Assistant Professor (CSE), ITER
   `sumitkumar@soa.ac.in`

   (9) Mr. Swadhin Kumar Barisal, Assistant Professor (CSE), ITER
   `swadhinbarisal@soa.ac.in`

   (10) Dr. Krishnendu Ghosh, Assistant Professor (CSE), ITER
   `krishnendughosh@soa.ac.in`

   (11) Dr. Prithviraj Mohanty, Assistant Professor (CSIT), ITER
   `prithvirajmohanty@soa.ac.in`

(12) Ms. Swarnima Singh Gautam, Assistant Professor (CSE), ITER
`swarnimasinghgautam@soa.ac.in`

5. **Text Book and References:**

    (1) Michael L. Scott, ***Programming Language Pragmatics***, 3rd Edition, ELSEVIER.

6. **Specific Course Information:**

    (a) **Course Description:** This course explores the principles, algorithms, and data structures involved in the design and construction of compilers. The course is intended to teach the students the basic techniques that underlie the practice of Compiler Construction. The course will introduce the theory and tools that can be standardly employed in order to perform syntax-directed translation of a high-level programming language into an executable code. Topics include finite-state machines, lexical analysis, syntax analysis, symbol tables, error recovery, intermediate code generation, code generation and optimization. These techniques can also be employed in wider areas of application, whenever we need a syntax-directed analysis of symbolic expressions and languages and their translation into a lower-level description. They have multiple applications for man-machine interaction, including verification and program analysis. In addition to the exposition of techniques for compilation, the course will also discuss various aspects of the run-time environment into which the high-level code is translated. This will provide deeper insights into the more advanced semantics aspects of programming languages, such as recursion, dynamic memory allocation, types and their inferences, object orientation, concurrency and multi-threading.

    (b) **Prerequisites and/or Co-requisites:**
    - ☞ CSE 3031 Theory of Computation.

7. **Course Learning Outcomes:**
By the end of course through lectures, readings, homeworks, assignments and exams, students will be able to:

    1. apply the principles in the theory of computation to the various stages in the design of compilers.
    2. explain the stages involved in the translation process.
    3. acquire knowledge in different phases and passes of Compiler, and specifying different types of tokens by lexical analyzer, and also able to use the compiler tools like LEX, YACC, etc.
    4. understand and design Parser(s) (LL, SLR, CLR and LALR) and its types i.e. Top-down and Bottom-up parsers.
    5. apply and evaluate syntax directed translation schemes, synthesized attributes, inherited attributes, different techniques for code optimization, symbol table organization, code generation and the fundamentals of run time environment.
    6. design different types of compiler tools to meet the requirements of the realistic constraints of compilers.

8. **Brief List of Topics to Be Covered: (L: Lecture, PSS: Problem Solving Session)**

Contact Hour	*Topics Covered*	*Remarks (if any)*
Week #1:		
L-0	Introduce the grading pattern, credit, classes and problem solving session of the course. Motivation behind the course. Introduction: The art of language design, the programming language spectrum, why study programming language.	**(Class: w.e.f 00th March 2022)** Academic regulation 2019/2020, Scott page 14-16
L-1	Introduction: Compilation and Interpretation, programming environments.	Scott page 16-24
L-2	**An overview of compilation**: Lexical and syntax analysis, Semantic analysis and intermediate code generation.	Scott page 25-29
PSS-1		
Week #2:		
L-3	**An overview of compilation**: target code generation, code improvement.	Scott page 33-35
L-4	**Programming Language Syntax**: Tokens and regular expressions.	Scott page 42-43
L-5	Context-free grammars, Derivations and parse trees.	Scott page 46-48
PSS-2	Automaton Design Using JFLAP Tool	
Week #3:		
L-6	Scanning: Generating a finite automaton.	Scott page 51-59
L-7	Scanner code.	Scott page 60-62
L-8	Scanner code, Table driven scanning.	Scott page 60-63
PSS-3		
Week #4:		
L-9	Lexical errors, pragams.	Scott page 63-65
L-10	Syntax Analysis: The role of parser.	Scott page 67-70
L-11	Recursive descent parsing.	Scott page 70-76
PSS-4	Programming using LEX	

Week #5:		
L-12	Recursive descent parsing contd ···	Scott page 70-76
L-13	Table-Driven Top-down parsing.	Scott page 76-87
L-14	Table-Driven Top-down parsing contd ···	Scott page 76-87
PSS-5	Programming using LEX (Contd...)	
Week #6:		
L-15	Table-Driven Top-down parsing contd ···	Scott page 76-87
L-16	Bottom-up parsing.	Scott page 87-99
L-17	Bottom-up parsing contd ···	Scott page 87-99
PSS-6	Programming using LEX (Contd...)	
Week #7:		
L-18	Bottom-up parsing contd ···	Scott page 87-99
L-19	Syntax errors.	Scott page 99 and Scott CD 1
L-20	**Names, Scopes, and binding**: The notion of binding time, object lifetime and storage management.	Scott page 111-118
PSS-7		
Week #8:		
L-21	**Scope rules**: Static scoping, nested subroutines, declaration order, modules, module types and classes, dynamic scoping.	Scott page 121-136
L-22	Implementing scopes, the meaning of names within a scope.	Scott page 143-148 and Scott CD 29,33
L-23	The binding of reference environments.	Scott page 151-157
PSS-8	Programming using YACC	
Week #9:		
L-24	Macro expansion, separate compilation.	Scott page 159-162 and Scott CD section
L-25	Chapter-3 exercises.	Scott page 163-171

L-26	**Semantic Analysis**: The role of the semantic analyser.	Scott page 175-176
PSS-9	Programming using YACC (Contd...)	
Week #10:		
L-27	Attribute grammars.	Scott page 180-182
L-28	Evaluating attributes.	Scott page 182-191
L-29	Action routines.	Scott page 191-196
PSS-10		
Week #11:		
L-30	**Space management for attributes**: Bottom-up evaluation.	Scott CD 49
L-31	**Space management for attributes**: Top-down evaluation.	Scott CD 54
L-32	Decorating a syntax tree, Chapter-4 exercise.	Scott page 197-204
PSS-11		
Week #12:		
L-33	**Control Flow**: Expression Evaluation, Structured and Unstructured Flow, Sequencing, Selection.	Scott page 219-255
L-34	Iteration, Recursion, Nondeterminacy, Chapter exercise.	Scott page 256-285 and Scott CD section
L-35	**Data Types**: Type Systems, Type Checking, Records (Structures) and Variants (Unions), Arrays, Strings, Sets, Pointers and Recursive Types, Lists, Equality Testing and Assignment.	Scott page 289-370
PSS-12		
Week #13:		
L-36	Subroutines and Control Abstraction, Review of Stack Layout, Calling Sequences, Parameter passing.	Scott page 383-409 and Scott CD section

L-37	**Data Abstraction and Object Orientation**: Object-Oriented Programming, Encapsulation and Inheritance, Initialization and Finalization, Dynamic Method Binding, Multiple Inheritance, exercise.	Scott page 449-491
L-38	**Building a Runnable Program**: Back-End Compiler Structure, Intermediate Forms, Code Generation, Address Space Organization, Assembly, Linking, Dynamic Linking.	Scott page 729-754 and Scott CD section
PSS-13		
Week #14:		
L-39	**Run-time Program Management**: Virtual Machines, Late Binding of Machine Code, Inspection/Introspection, chapter exercise.	Scott page 761-812
L-40	**Code Improvement**: Phases of Code Improvement, Peephole Optimization, Redundancy Elimination in Basic Blocks.	Scott CD section
L-41	Global Redundancy and Data Flow Analysis, Loop Improvement I, Instruction Scheduling, Loop Improvement II, Register Allocation , Chapter exercise.	Scott CD section
PSS-14		

9. **Evaluation scheme (under Grading Pattern-1):**

Attendance	:	05%
Assignments / Quizes	:	20%
Mid-Term Examination	:	15%
PSS Examination	:	15%
End-Term Examination	:	45%

10. **Program Outcomes & Program Specific Outcomes:**
There are twelve program outcomes (1-12) for the Computer science & Engineering B. Tech program:
Program Outcomes (POs)

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specifis Outcomes (PSOs)

PSO 1. The ability to understand, analyze and develop computer programs in the areas related to business intelligence, web design and networking for efficient design of computer-based systems of varying complexities.

PSO 2. The ability to apply standard practices and strategies in software development using open-ended programming environments to deliver a quality product for business success.

11. **Correlation between the Course Outcomes(COs), the Program Outcomes(POs), and the Program Specific Outcomes(PSOs)**

Course Articulation Matrix:

Programming Language and Compilers (CSE 4021)													Session: 2021-2022	
	POs												PSOs	
	1	2	3	4	5	6	7	8	9	10	11	12	1	2
CO 1	x													
CO 2		x												
CO 3			x		x									
CO 4			x											
CO 5				x										
CO 6			x		x									
Average	x	x	x	x	x									