J.B. INSTITUTE OF ENGINEERING & TECHNOLOGY

(AUTONOMOUS)

COURSE FILE OF

COMPILER DESIGN

III B.TECH	II SEMESTER	ACADEMIC YEAR	2015-16

D HIMAGIRI

ASSISTANT PROFESSOR,

CSE DEPARTMENT

http://www.jbiet.edu.in

	COURSE PLAN	2015-16	
		12	Regulation: R

FACULTY DETAILS:

Name of the Faculty:: HIMAGIRI DANAPANA
Designation: ASSISTANT PROFESSOR
Department:: CSE

COURSE DETAILS

Name Of The Programme::	B.Tech	Batch::	13-17
Designation::	Asst.Professor		
Year	III	Semester	II
Department::	CSE		
Title of The Subject	COMPILER DESIGN	Subject Code	6756031
No of Students	137		

	COURSE PLAN	2015-16 Regulation: R12

FACULTY DETAILS:

Name of the Faculty:: HIMAGIRI DANAPANA
Designation: ASSISTANT PROFESSOR
Department:: CSE

1. TARGET

a) Percentage Pass 80

b) Percentage I class 60

2. COURSE PLAN

(Please write how you intend to cover the contents: i.e., coverage of Units by lectures, guest lectures, design exercises, solving numerical problems, demonstration of models, model preparation, or by assignments, etc.)

A. COVERAGE OF UNITS BY LECTURES B. Solving numerical problems
C. ASSIGNMENTS D. DESIGN EXERCISES

3. METHOD OF EVALUATION

3.1. □ Continuous Assessment Examinations (CAE 1, CAE 2)

3.2. □ Assignments / Seminars

3.3. □ Mini Projects

3.4. □ Quiz

3.5. □ Term End Examination

3.6. □ Others

4. List out any new topic(s) or any innovation you would like to introduce in teaching the subject in this Semester.

Parallel architecture based compilers

Signature of HOD
Date:

Signature of Faculty
Date:

	GUIDELINES TO STUDY THE SUBJECT	2015-16 Regulation: R12

FACULTY DETAILS:

Name of the Faculty::	HIMAGIRI DANAPANA
Designation:	ASSISTANT PROFESSOR
Department::	CSE
Guidelines for Preparing the Course:	

Course Description:

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 employed in order to perform syntax-directed translation of a high-level programming language into an executable code.

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.

Course Objectives:

1. Compiler writing is a basic element of programming language research. Many language researchers write compilers for the languages they design.
2. Many applications have similar properties to one or more phases of a compiler, and compiler expertise and tools can help an application programmer working on other projects besides compilers.

Learning Outcomes:

1. Writing a compiler gives a student experience with large-scale applications development. Your compiler program may be the largest program you write as a student. Experience working with really big data structures and complex interactions between algorithms will help you out on your next big programming project.

2. Compiler writing is one of the shining triumphs of CS theory. It demonstrates the value of theory over the impulse to just "hack up" a solution.

	COURSE OBJECTIVES	2015-16 Regulation: R12

FACULTY DETAILS:

Name of the Faculty:: HIMAGIRI DANAPANA
Designation: ASSISTANT PROFESSOR
Department:: CSE

On completion of this Subject / Course the student shall be able to:

S.No.	Objectives	Outcomes
1. 2	Define the Phases of a Compiles Perform lexical analysis	Knowledge/Define Application of knowledge/Demonstrate
3	Use lexical analyzer generators	Application of knowledge/Demonstrate
4 6	Read and write grammars for programming language constructs Define ambiguous grammars,	Understand/Describe Knowledge/Define
7	Define LL(1),	Knowledge/Define
8	Perform Top-Down Parsing ,	Understand/Give Examples
9	Use top-down parser generators	Application of knowledge/Demonstrate
10 11	Perform Bottom-Up Parsing Define SLR(1), LR, LALR	Understand/Give Examples Knowledge/Define , Understand/Give Examples
12	Use Bottom-up parser generators	Apply/ Illustrate
13	Perform semantic analysis	Apply/ Solve
14	Define static checking	Knowledge/Define
15 16	Define intermediate representations Read and write attribute grammars	Knowledge/Define Understand/Describe
17 18	Perform local, loop and global optimization Discover Basic Blocks	Knowledge/Define , Understand/Give Examples Apply/ Solve
19 20 21	Create a Control Flow Graph, Discover Loops Define Data Flow Problems Solve Data Flow Problem	Apply/ Illustrate Knowledge/Define Apply/ Solve

Signature of Faculty
Date:

Note: For each of the OBJECTIVE indicate the appropriate OUTCOMES to be achieved.
Kindly refer Page 16, to know the illustrative verbs that can be used to state the objectives.

	COURSE OUTCOMES	2015-16
		Regulation: R12

FACULTY DETAILS:

Name of the Faculty:: HIMAGIRI DANAPANA
Designation: ASSISTANT PROFESSOR
Department:: CSE

The expected outcomes of the Course / Subject are:

S.No.	General Categories of Outcomes	Specific Outcomes of the Course
A.	An ability to apply knowledge of mathematics, science, and engineering	Compute, Solve
B.	An ability to design and conduct experiments, as well as to analyze and interpret data	Design, Devise
C.	An ability to design a system, component, or process to meet desired needs within realistic Constraints such as economic, environmental, social, political, ethical, health and safety, Manufacturability and sustainability	
D.	An ability to function on multi-disciplinary teams	
E.	An ability to identify, formulate, and solve engineering problems	Estimate
F.	An understanding of professional and ethical responsibility	Knowledge
G.	An ability to communicate effectively	Relate
H.	The broad education necessary to understand the impact of engineering solutions in a global, economic, environmental, and societal context	Prepare
I.	A recognition of the need for, and an ability to engage in life-long learning	Appraise
J.	A knowledge of contemporary issues	
K.	An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.	Appraise

Objectives – Outcome Relationship Matrix (Indicate the relationships by ☒ mark)**.**

Outcomes Objectives	A	B	C	D	E	F	G	H	I	J	K
1.	□	□	□	□	□	□	□	□	□	□	□
2.	□	□	□	□	□	□	□	□	□	□	□
3.	□	□	□	□	□	□	□	□	□	□	□
4.	□	□	□	□	□	□	□	□	□	□	□
5.	□	□	□	□	□	□	□	□	□	□	□
6.	□	□	□	□	□	□	□	□	□	□	□
7.	□	□	□	□	□	□	□	□	□	□	□
8.	□	□	□	□	□	□	□	□	□	□	□
9.	□	□	□	□	□	□	□	□	□	□	□
10.	□	□	□	□	□	□	□	□	□	□	□

COURSE SCHEDULE

2015-16

Regulation: R12

FACULTY DETAILS:

Name of the Faculty:: HIMAGIRI DANAPANA
Designation: ASSISTANT PROFESSOR
Department:: CSE
The Schedule for the whole Course / Subject is:: COMPILER DESIGN

S. No.	Description	Duration (Date)		Total No. of Periods
		From	To	
1.	Overview of Compilation	14/12/2015	29/1/2016	10
2.	Top down Parsing	30/1/2016	12/1/2016	10
3.	Bottom up parsing	12/1/2016	1/2/2016	10
4.	Semantic analysis	3/2/2016	6/2/2016	9
5.	Symbol tables	15/2/2016	25/2/2016	10
6.	Code optimization	26/2/2016	10/3/2016	8
7	Data flow analysis	11/3/2016	/26/2016	10
8	Object code generation	27/3/2016	9/4/2016	8

Total No. of Instructional periods available for the course: Hours / Periods

	SCHEDULE OF INSTRUCTIONS **UNIT - I**	2015-16 Regulation: R12

FACULTY DETAILS:

Name of the Faculty:: HIMAGIRI DANAPANA
Designation: ASSISTANT PROFESSOR
Department:: CSE
The Schedule for the whole Course / Subject is:: COMPILER DESIGN

Sl. No.	Date	No. of Periods	Topics / Sub - Topics	Objectives & Outcome Nos.	References (Text Book, Journal…) Page No___ to ___
1.	14/12/2015	3	Phases of Compilation – Lexical Analysis	1 & 2,A&B	TB1-1.2,TB2-1.3
2	15/12/2015 – 20/12/2016	3	Regular Grammar and regular expression for common programming language features	1, A&B	TB1-3.3,TB2-1.9&2.1
3	21/12/2015	1	Pass and Phases of translation	1, F	TB-1 1.2.8
4	23/12/2016	1	Interpretation ,Bootstrapping	1,F	LINK
5	28 - 29/12/2015	2	Data structures in compilation – LEX lexical analyzer generator	1,2 & B	TB-1 3.5

Signature of Faculty
Date

Note: 1. ENSURE THAT ALL TOPICS SPECIFIED IN THE COURSE ARE MENTIONED.
2. ADDITIONAL TOPICS COVERED, IF ANY, MAY ALSO BE SPECIFIED **BOLDLY**.
3. MENTION THE CORRESPONDING COURSE OBJECTIVE AND OUT COME NUMBERS AGAINST EACH TOPIC.

	SCHEDULE OF INSTRUCTIONS **UNIT - II**	2015-16 Regulation: R12

FACULTY DETAILS:

Name of the Faculty:: HIMAGIRI DANAPANA
Designation: ASSISTANT PROFESSOR
Department:: CSE
The Schedule for the whole Course / Subject is:: COMPILER DESIGN

Sl. No.	Date	No. of Periods	Topics / Sub - Topics	Objectives & Outcome Nos.	References (Text Book, Journal…) Page No___ to ___
1	30/12/2015-1/1/2016	2	Context free grammars	4,6 &A	TB –1 4.2
2	2/1/2016	1	Top down parsing – Backtracking	8 & B	TB –1 4.4
3	4/1/2016 – 6/1/2016	3	LL (1)	7 & B	TB –1 4.4.3
4	8/1/2016-9/1/2016	2	Recursive descent parsing	8 & A	TB –1 4.4.1
5	11/1/2016	1	Predictive parsing	8 & A	TB –1 4.4.4
6	12/1/2016	1	Pre-processing steps required for predictive parsing	6, A & B	TB-1 4.4.5

Signature of Faculty
Date

Note: 1. ENSURE THAT ALL TOPICS SPECIFIED IN THE COURSE ARE MENTIONED.
2. ADDITIONAL TOPICS COVERED, IF ANY, MAY ALSO BE SPECIFIED **BOLDLY**.
MENTION THE CORRESPONDING COURSE OBJECTIVE AND OUT COME NUMBERS AGAINST EACH TOPIC.

	SCHEDULE OF INSTRUCTIONS **UNIT - III**	2015-16 Regulation: R12

FACULTY DETAILS:

Name of the Faculty:: HIMAGIRI DANAPANA
Designation: ASSISTANT PROFESSOR
Department:: CSE
The Schedule for the whole Course / Subject is:: COMPILER DESIGN

Sl. No.	Date	No. of Periods	Topics / Sub - Topics	Objectives & Outcome Nos.	References (Text Book, Journal…) Page No___ to ___
1	12/1/206- 20/1/2016	3	Shift Reduce parsing	11 & A	TB-1 4.5.3
2	22/1/2016- 23/1/2016	3	LR and LALR parsing	11 A	TB-1 4.6
3	25/1/2016	1	Error recovery in parsing	11 E	TB-1 4.8.3
4	27/1/2016	1	handling ambiguous grammar	6 A	TB-1 4.8
5	27/1/2016- 29/1/2016	2	YACC – automatic parser generator	12 K	TB-1 4.9.2

Signature of Faculty
Date

Note: 1. ENSURE THAT ALL TOPICS SPECIFIED IN THE COURSE ARE MENTIONED.

2. ADDITIONAL TOPICS COVERED, IF ANY, MAY ALSO BE SPECIFIED **BOLDLY**.

MENTION THE CORRESPONDING COURSE OBJECTIVE AND OUT COME NUMBERS AGAINST EACH TOPIC.

	SCHEDULE OF INSTRUCTIONS **UNIT - IV**	2015-16 Regulation: R12

FACULTY DETAILS:

Name of the Faculty:: HIMAGIRI DANAPANA
Designation: ASSISTANT PROFESSOR
Department:: CSE
The Schedule for the whole Course / Subject is:: COMPILER DESIGN

Sl. No.	Date	No. of Periods	Topics / Sub - Topics	Objectives & Outcome Nos.	References (Text Book, Journal…) Page No___ to ___
1	3/2/2016	2	Intermediate forms of source Programs – abstract syntax tree	15 & A	TB1-2.8&5.3,TB2-1.2,3
2	4/2/2016	2	Polish notation and three address codes	15 & A	TB1-5.1,TB2-3.1
3	5/2/2016	2	Attributed grammars	16 & E	TB1-5.2,TB2-3
4	5/2/2016	1	Syntax directed translation	16 & E	TB1-5.4,TB2-3
5	6/2/2016	1	Conversion of popular Programming languages	16 & F	TB1-6,TB2-6.4,6.5
6	6/2/2016	1	Language Constructs into Intermediate code forms,typechecker	16 & F	TB-1 6.4

Signature of Faculty
Date

Note: 1. ENSURE THAT ALL TOPICS SPECIFIED IN THE COURSE ARE MENTIONED.
2. ADDITIONAL TOPICS COVERED, IF ANY, MAY ALSO BE SPECIFIED **BOLDLY**.
MENTION THE CORRESPONDING COURSE OBJECTIVE AND OUT COME NUMBERS AGAINST EACH TOPIC.

	SCHEDULE OF INSTRUCTIONS **UNIT - V**	2015-16 Regulation: R12

FACULTY DETAILS:

Name of the Faculty:: HIMAGIRI DANAPANA
Designation: ASSISTANT PROFESSOR
Department:: CSE
The Schedule for the whole Course / Subject is:: COMPILER DESIGN

Sl. No.	Date	No. of Periods	Topics / Sub - Topics	Objectives & Outcome Nos.	References (Text Book, Journal…) Page No___ to ___
1	15/2/2016	1	Symbol table format		TB1-2.7,TB2-2.1
2	16/2/2016	1	Organization for block structures languages		TB1-1.6
3	18/2/2016	1	Hashing		TB1-2,TB2-2.1
4	20/2/2016	1	Tree structures representation of scope information		TB1-2.7,TB2-2.1
5	21/2/2016,22/2/2016	2	Block structures and non block structure storage allocation: • static, • Runtime stack • heap storage allocation		B1-7.1,TB2-6
6	23/2/204,25/2/2016	2	Storage allocation for arrays, strings and records.		TB1-7.1,TB2-6

Signature of Faculty
Date

Note: 1. ENSURE THAT ALL TOPICS SPECIFIED IN THE COURSE ARE MENTIONED.
2. ADDITIONAL TOPICS COVERED, IF ANY, MAY ALSO BE SPECIFIED **BOLDLY**.
MENTION THE CORRESPONDING COURSE OBJECTIVE AND OUT COME NUMBERS AGAINST EACH TOPIC.

	SCHEDULE OF INSTRUCTIONS **UNIT - VI**	2015-16 Regulation: R12

FACULTY DETAILS:

Name of the Faculty:: HIMAGIRI DANAPANA
Designation: ASSISTANT PROFESSOR
Department:: CSE
The Schedule for the whole Course / Subject is:: COMPILER DESIGN

SI. No.	Date	No. of Periods	Topics / Sub - Topics	Objectives & Outcome Nos.	References (Text Book, Journal…) Page No___ to ___
1	26/2/2016	1	Consideration for Optimization		TB1-8,9
2	27/2/2016	1	Scope of Optimization		TB1-8.4
3	29/2/2016,30/2/2016	2	local optimization		TB-1 9.7
4	2/3/2016-3/3/2016	1	loop optimization		TB-1 9.7
5	5/3/2016	1	frequency reduction		TB1-6.1
7	8/3/2016	1	Folding		TB-1 8.5
8	10/3/2016	1	DAG representation		TB-1 9.6

Signature of Faculty
Date

Note: 1. ENSURE THAT ALL TOPICS SPECIFIED IN THE COURSE ARE MENTIONED.
2. ADDITIONAL TOPICS COVERED, IF ANY, MAY ALSO BE SPECIFIED **BOLDLY**.
MENTION THE CORRESPONDING COURSE OBJECTIVE AND OUT COME NUMBERS AGAINST EACH TOPIC.

	SCHEDULE OF INSTRUCTIONS **UNIT - VII**	2015-16 Regulation: R12

FACULTY DETAILS:

Name of the Faculty:: HIMAGIRI DANAPANA
Designation: ASSISTANT PROFESSOR
Department:: CSE
The Schedule for the whole Course / Subject is:: COMPILER DESIGN

Sl. No.	Date	No. of Periods	Topics / Sub - Topics	Objectives & Outcome Nos.	References (Text Book, Journal…) Page No___ to ___
1	11/3/2016	1	Flow graph		TB-1 9.1
2	13-14/3/2016	2	Data flow equation		TB1-9.2,RB4-17
3	15/3/2016	1	Global optimization		TB1-9.2,RB4-17
4	16-17/3/2016	2	Redundant sub expression elimination		TB1-9.2,RB4-17
5	18-20/3/2016	2	Induction variable elements		RB4-18, TB1-9.1
6	22/3/2016	1	Live variable analysis		RB4-17 ,TB1-9.2
7	26/3/2016	**1**	Copy propagation		TB-2 6.7

Signature of Faculty
Date

Note: 1. ENSURE THAT ALL TOPICS SPECIFIED IN THE COURSE ARE MENTIONED.
2. ADDITIONAL TOPICS COVERED, IF ANY, MAY ALSO BE SPECIFIED **BOLDLY.**
MENTION THE CORRESPONDING COURSE OBJECTIVE AND OUT COME NUMBERS AGAINST EACH TOPIC.

	SCHEDULE OF INSTRUCTIONS **UNIT - VIII**	2015-16 Regulation: R12

FACULTY DETAILS:

Name of the Faculty:: HIMAGIRI DANAPANA
Designation: ASSISTANT PROFESSOR
Department:: CSE
The Schedule for the whole Course / Subject is:: COMPILER DESIGN

Sl. No.	Date	No. of Periods	Topics / Sub - Topics	Objectives & Outcome Nos.	References (Text Book, Journal…) Page No___ to ___
1	27-28/3/2016	2	Object code forms		TB-2 9.1
2	29/3/2016,30/3/2016	2	Machine dependent code optimization		TB-2 9.2
3	03/4/2016-04/4/2016	2	register allocation and assignment generic code generation algorithms		TB-1 8.6.2
4	08-09/4/2016	2	DAG for register allocation.		TB-1 8.5.1

Signature of Faculty
Date

Note: 1. ENSURE THAT ALL TOPICS SPECIFIED IN THE COURSE ARE MENTIONED.
2. ADDITIONAL TOPICS COVERED, IF ANY, MAY ALSO BE SPECIFIED **BOLDLY**.
MENTION THE CORRESPONDING COURSE OBJECTIVE AND OUT COME NUMBERS AGAINST EACH TOPIC.

	COURSE COMPLETION STATUS	2015-16
		Regulation: R12

FACULTY DETAILS:

Name of the Faculty:: HIMAGIRI DANAPANA

Subject:: COMPILER DESIGN Subject Code : 6756031

Department:: CSE

Actual Date of Completion & Remarks, if any

Units	**Remarks**	**Nos. of Objectives Achieved**
Unit 1	Learnt about lexical analysis	3
Unit 2	Leant about Top Down parsing	5
Unit 3	Learnt about Bottom – Up parsing	3
Unit 4	Learnt about semantic analysis	2
Unit 5	Leant about intermediate representations	2
Unit 6	Learn about code optimization	2
Unit 7	Learn about code generation	2
Unit 8	Learn about data flow diagrams	2

Signature of Dean of School
Date:

Signature of Faculty
Date:

NOTE: AFTER THE COMPLETION OF EACH UNIT MENTION THE NUMBER OF OBJECTIVES ACHIEVED.

	# TUTORIAL SHEETS - I	2015-16
		Regulation: R12

FACULTY DETAILS:

Name of the Faculty::	HIMAGIRI DANAPANA
Designation:	ASSISTANT PROFESSOR
Department::	CSE
The Schedule for the whole Course / Subject is::	COMPILER DESIGN

Date:

This Tutorial corresponds to Unit Nos.1, 2

Time:

Q1. Describe various phases of a compiler? Differentiate a phase and pass? Compare multipass and single pass compiler?

Q2. Explain with one example how LEX program perform lexical analysis for the following patterns in `C': identifier, comments, numerical constants, arithmetic operators.

Q3. .(a) Construct an NFA for regular expression R= (aa b)*ab convert it into an equivalent DFA

(b). Write short notes on bootstrapping process.

Q4. (a) Construct recursive descent parser for the following grammar.

E → T E'
E' → +T E'|e
T → F T'
T' → *FT'|e
F → (E)|id

(b) Eliminate ambiguities in the following grammar.

S → iEtS| iEtSeS| a
E → b| c | d

Q5. (a) What are the difficulties in top down parsing? Explain in detail.

(b) Consider the following grammar

S → (L) |a
L → L, S |S

Construct leftmost derivations and parse trees for the following sentences:

i. (a,(a,a))
ii. (a,((a,a),(a,a))).

Please write the Questions / Problems / Exercises which you would like to give to the students and also mention the objectives to which these questions / Problems are related.

Signature of Dean of School
Date:

Signature of Faculty
Date:

	TUTORIAL SHEETS - II	2015-16
		Regulation: R12

FACULTY DETAILS:

Name of the Faculty:: HIMAGIRI DANAPANA
Designation: ASSISTANT PROFESSOR
Department:: CSE
The Schedule for the whole Course / Subject is:: COMPILER DESIGN

Date:

This Tutorial corresponds to Unit Nos.3, 4, 5

Time:

Q1. .a) Give the rules for computation of FIRST(X) and FOLLOW(X). Construct

FIRST and FOLLOW sets for the following grammar.

E → T E'
E' → +T E'|e
T → F T'
T' → *FT'|e
F → (E)|id

b) Construct SLR parsing table for following grammar:

E → E + T/T
T → T * F/F
F → (E)/id.

Q2. Construct Canonical LR parsing table for the following grammar

S → CC
C → cC/d

Q3. Construct LALR parsing table for the following grammar

S →CC
C -→cC/d

Q4. (a) Write the quadruples, triples and indirect triples for the expression

i. (a + b) *(c + d) *(a + b + c)
ii. a * (b + c)

(b). Draw syntax tree for the arithmetic expressions

A*(b + c) − d/2

Also write the given expression in postfix notation.

Q5. (a) Write a short note on L-attributed grammars.

(b) It is required to compute the total number of reductions performed to parse a given input. Using synthesized attributes only write the semantic rules to find E.red, the number of reductions performed while reducing on input to E.

E → E * T/T
T → F - T/F
F →2/4

Also draw annotated tree for 4-2-4*2.

Please write the Questions / Problems / Exercises which you would like to give to the students and also mention the objectives to which these questions / Problems are related.

Signature of Dean of School
Date:

Signature of Faculty
Date:

	TUTORIAL SHEETS - II	2015-16 Regulation: R12

FACULTY DETAILS:

Name of the Faculty:: HIMAGIRI DANAPANA
Designation: ASSISTANT PROFESSOR
Department:: CSE

Date:

This Tutorial corresponds to Unit Nos.6,7,8

Time:

Q1. (a) What is an ordered and unordered symbol table? What is the function of symbol table in the compilation process? Explain.

(b) What are the various attributes of a Symbol Table?

Q2. Explain DAG and its use. Write the procedure to construct the DAG for a statement.

Q3.(a) Explain different principal sources of optimization technique with suitable examples.

b) What is DAG? Construct the DAG for the following basic block

```
D := B_C
E :=A+B
B := B+C
A := E-D
```

Q4. Describe, how redundant expression elimination can be done in loop optimization technique, during global optimization.

Q5. a) Explain the concept of object code forms.

(b) Generate optimal machine code for the following C program.

```
main()
{
int i, a[10];
while (i<=10) a[i] =0
}
```

Please write the Questions / Problems / Exercises which you would like to give to the students and also mention the objectives to which these questions / Problems are related.

Signature of Dean of School
Date:

Signature of Faculty
Date:

	ILLUSTRATIVE VERBS FOR STATING INSTRUCTIONAL OBJECTIVES	2015-16 Regulation: R12

These verbs can also be used while framing questions for Continuous Assessment Examinations as well as for End – Semester (final) Examinations.

ILLUSTRATIVE VERBS FOR STATING **GENERAL OBJECTIVES**

Know Comprehend	Understand Apply	Analyze Design	Generate Evaluate

ILLUSTRATIVE VERBS FOR STATING **SPECIFIC OBJECTIVES:**

A. Cognitive Domain

1	2	3	4	5	6
Knowledge	**Comprehension Understanding**	**Application** of knowledge & comprehension	**Analysis** of whole w.r.t. its constituents	**Synthesis** combination of ideas/constituents	**Evaluation** judgement
Define Identify Label List Match Reproduce Select State	Convert Defend Describe (a procedure) Distinguish Estimate Explain why/how Extend Generalize Give examples Illustrate Infer Summarize	Change Compute Demonstrate Deduce Manipulate Modify Predict Prepare Relate Show Solve	Breakdown Differentiate Discriminate Distinguish Separate Subdivide	Categorize Combine Compile Compose Create Devise Design Generate Organize Plan Rearrange Reconstruct Reorganize Revise	Appraise Compare Conclude Contrast Criticize Justify Interpret Support

B. Affective Domain

Adhere
Assist
Attend
Change
Develop
Help
Influence
Initiate
Resolve
Select
Serve
Share

C. Psychomotor Domain (skill development)

Bend
Calibrate
Compress
Conduct
Connect
Convert
Decrease
Demonstrate
Dissect
Draw
Extend
Feed
File
Grow
Handle
Increase
Insert
Keep
Elongate
Limit
Manipulate
Move precisely
Operate
Paint
Perform
Prepare
Remove
Replace
Report
Reset
Run
Set
Straighten
Strengthen
Time
Transfer
Type
Weigh

	LESSON PLAN **Unit-1**	2015-16 Regulation: R12

Name of the Faculty:	HIMAGIRI DANAPANA		
Subject	COMPILER DESIGN	Subject Code 6756031	
Unit	I		

INSTRUCTIONAL OBJECTIVES:

Session No	Topics to be covered	Time	Ref	Teaching Method
1	Phases of Compilation – Lexical Analysis	50 min	TB1-1.2,TB2-1.3	Chalk& board ,charts
2	Regular Grammar and regular expression for common programming language features,	50 min	TB1-3.3,TB2-1.9&2.1	Chalk& board ,charts
3	Pass and Phases of translation	50 min	TB-1 1.2.8	Chalk& board ,charts
4	Interpretation	50 min	LINK	Chalk& board ,charts
5	Bootstrapping	50 min	LINK	Chalk& board ,charts
6	Data structures in compilation – LEX lexical analyzer generator.	50 min	TB-1 3.5	Chalk& board ,charts
7				
8				
9				
10				

On completion of this lesson the student shall be able to(Outcomes)

1.

2.

3.

4

	ASSIGNMENT **Unit-I**	2015-16
		Regulation: R12

Assignment / Questions

1. (a) Explain, in detail, lexical analyzer generator.
(b) Describe the lexical errors and various error recovery strategies with suitable examples.

2. (a) Consider the following fragment of ‘C’ code:
float i, j;
i = i * 70 + j + 2;
.Write the output at all phases of the compiler for the above ‘C’ code.
(b). Explain the input buffer scheme for scanning the source program. How the use of sentinels can improve its performance? Describe in detail.

3.(a) Write a regular expressions and NFA for the following patterns. Use auxiliary definitions where convenient?
i. The set of words having a, e, i, o, u appearing in that order, although not having necessarily consecutively.
ii. Comments in C.
(b) Differentiate Interpreter & Compiler?

4. Describe various phases of a compiler? Differentiate a phase and pass? Compare multi-pass and single pass compiler?

5. Explain with one example how LEX program perform lexical analysis for the following patterns in `C': identifier, comments, numerical constants, arithmetic operators.
(b) Write short notes on bootstrapping process.

Signature of Faculty

Note: Mention for each question the relevant objectives and outcomes.

	LESSON PLAN **Unit-II**	2015-16 Regulation: R12

Name of the Faculty: HIMAGIRI DANAPANA

Subject COMPILER DESIGN Subject Code 6756031

Unit **II**

INSTRUCTIONAL OBJECTIVES:

Session No	**Topics to be covered**	**Time**	**Ref**	**Teaching Method**
1	Context free grammars	**50 min**	TB –1 4.2	Chalk& board ,PPTs, excercises
2	Top down parsing – Backtracking	**50 min**	TB –1 4.4	Chalk& board ,PPTs, excercises
3	LL (1)	**50 min**	TB –1 4.4.3	Chalk& board ,PPTs, excercises
4	Recursive descent parsing	**50 min**	TB –1 4.4.1	Chalk& board ,PPTs, excercises
5	Predictive parsing	**50 min**	TB –1 4.4.4	Chalk& board ,PPTs, excercises
6.	Pre-processing steps required for predictive parsing	**50 min**	TB-1 4.4.5	Chalk& board ,PPTs, excercises

On completion of this lesson the student shall be able to

1.

2.

3.

4

	ASSIGNMENT **Unit-II**	2015-16 Regulation: R12

Assignment / Questions

1. (a) Construct recursive descent parser for the following grammar.
E → T E’
E’ → +T E’|e
T → F T’
T’ → *FT’|e
F → (E)|id
(b) Eliminate ambiguities in the following grammar.
S → iEtS| iEtSeS| a
E → b| c | d

2. (a) What are the difficulties in top down parsing? Explain in detail.
(b) Consider the following grammar
S → (L) |a
L → L, S |S
Construct leftmost derivations and parse trees for the following sentences:
i. (a,(a,a))
ii. (a,((a,a),(a,a))).

3.Construct SLR parsing table for following grammar:
E → E + T/T
T → T * F/F
F → (E)/id.

4.(a) What is CFG? Explain in detail.
(b) Write short notes on following terms:
i. Parse tree.
ii. LL(k) grammar.

Signature of Faculty

Note: Mention for each question the relevant objectives and outcomes.

	LESSON PLAN **Unit-III**	2015-16
		Regulation: R12

Name of the Faculty: HIMAGIRI DANAPANA

Subject COMPILER DESIGN Subject Code 6756031

Unit **III**

INSTRUCTIONAL OBJECTIVES:

Session No	**Topics to be covered**	**Time**	**Ref**	**Teaching Method**
1	Shift Reduce parsing	**50 min**	TB-1 4.5.3	Chalk& board Exercises ,PPTs
2	LR and LALR parsing	**50 min**	TB-1 4.6	Chalk& board Exercises
3	Error recovery in parsing	**50 min**	TB-1 4.8.3	,PPTs
4	handling ambiguous grammar	**50 min**	TB-1 4.8	Chalk& board Exercises
5	YACC – automatic parser generator	**50 min**	TB-1 4.9.2	,PPTs

On completion of this lesson the student shall be able to(Outcomes)

1.

2.

3.

4

	ASSIGNMENT **Unit-III**	2015-16
		Regulation: R12

Assignment / Questions

1. (a) Explain the stack implementation of shift reduce parsing method with an example.
(b) Define handle. Give suitable example.

2. Construct Canonical LR parsing table for the following grammar
S → CC
C → cC/d

3. (a) What are the actions of shift-reduce parsers?
(b).Construct LALR parsing table for the following grammar.
A′ → A
A → (A)
A → a

4.(a) Construct LALR parsing table for the following grammar
S →CC
C →cC/d
(b) What do you mean by left most derivation? Explain with an example

5. Write the algorithm to generate the canonical collection of LR(0) items

6. (a) What is an SLR Grammar?
(b) Construct LALR(1) Parse table from the following grammar
S→ Aa/bAc/dc/bda:.

Signature of Faculty

Note: Mention for each question the relevant objectives and outcomes.

	LESSON PLAN **Unit-IV**	2015-16 Regulation: R12

Name of the Faculty:	HIMAGIRI DANAPANA		
Subject	COMPILER DESIGN	Subject Code	6756031
Unit	**IV**		

INSTRUCTIONAL OBJECTIVES:

Session No	**Topics to be covered**	**Time**	**Ref**	**Teaching Method**
1	Intermediate forms of source Programs – abstract syntax tree	**50 min**	TB1-2.8&5.3,TB2-1.2,3	Chalk& board Exercises, PPTs
2	Polish notation and three address codes	**50 min**	TB1-5.1,TB2-3.1	Chalk& board Exercises, PPTs
3	Attributed grammars	**50 min**	TB1-5.2,TB2-3	Chalk& board Exercises, PPTs
4	Syntax directed translation	**50 min**	TB1-5.4,TB2-3	Chalk& board Exercises, PPTs
5	Conversion of popular Programming languages	**50 min**	TB1-6,TB2-6.4,6.5	Chalk& board Exercises, PPTs
6	Language Constructs into Intermediate code forms	**50 min**	TB-1 6.4	Chalk& board Exercises, PPTs

On completion of this lesson the student shall be able to (Outcomes)

1.

2.

3.

4

	ASSIGNMENT **Unit-IV**	2015-16 Regulation: R12

Assignment / Questions

1. (a) Write a note on the specification of a simple type checker.
(b) What is a type expression? Explain the equivalence of type expressions with appropriate examples.

2.(a) Draw syntax tree for the arithmetic expressions **A*(b + c) − d/2**
Also write the given expression in postfix notation.
(b) Write the quadruple, triple, indirect triple for the following expression
(x + y)*(y + z) + (x + y + z)

3. (a) Describe the overloading of functions and operators with suitable examples.
(b) Write a note on polymorphic functions.

4. Write type expressions for the following types.
(a) An array of pointers to reals, where array index ranages from 1 to 100.
(b) A two dimensional array of integers (i.e. an array of array) whose rows are indexed from 0 to 9 and whose columns are indexed from -10 to 10.
(c) Functions whose domains are functions from integers to pointers to integers
and whose ranges are records consisting of an integer and a character

5.(a) Write the quadruples, triples and indirect triples for the expression
i. (a + b) *(c + d) *(a + b + c)
ii. a * (b + c)
1
S→ AS | b
A → SA | a
(b) Write a top-down translation scheme to produce quadruples for Boolean Ex-
Pression

6.a) Why are quadruples preferred over triples in an optimizing compiler. Explain
(b) Give the triple representation of an array operation x : = y [i]
(c) Give the syntax directed definition of if else statement.

Signature of Faculty

Note: Mention for each question the relevant objectives and outcomes.

	LESSON PLAN **Unit-V**	2015-16
		Regulation: R12

Name of the Faculty: HIMAGIRI DANAPANA

Subject COMPILER DESIGN Subject Code 6756031

Unit **V**

INSTRUCTIONAL OBJECTIVES:

Session No	**Topics to be covered**	**Time**	**Ref**	**Teaching Method**
1	Symbol table format	**50 min**	TB1-2.7,TB2-2.1	Chalk& board Exercises
2	Organization for block structures languages	**50 min**	TB1-1.6	Chalk& board Exercises
3	Hashing	**50 min**	TB1-2,TB2-2.1	Chalk& board Exercises
4	Tree structures representation of scope information	**50 min**	TB1-2.7,TB2-2.1	Chalk& board Exercises
5	Block structures and non block structure storage allocation: • static, • Runtime stack • heap storage allocation	**50 min**	B1-7.1,TB2-6	Chalk& board Exercises
6	Tree structures representation of scope information	**50 min**	TB1-7.1,TB2-6	Chalk& board Exercises, PPTs

On completion of this lesson the student shall be able to (Outcomes)

1.

2.

3.

4

	ASSIGNMENT **Unit-V**	2015-16 Regulation: R12

Assignment / Questions

1. (a) What is an ordered and unordered symbol table? What is the function of symbol table in the compilation process? Explain.
(b) What are the various attributes of a Symbol Table?

2.(a) Compare three different storage allocation strategies.
(b) Consider the following array declaration in ‘c’;
float a[100][100];
Assume that the main memory in byte addressable and that the array is stored starting from the memory address 100.
What is the address of a[40][50]

3.(a) Explain the hash table with temporary and permanent storage.
(b) Reusing the storage space for names.

4.Explain symbol table organization using hash tables? Construct hash based structure for symbol table for the variable in the following program.

```
        int main()
{
int a1, a2, c1, c2;
char b1;
float d1, d2;
}
```

Signature of Faculty

Note: Mention for each question the relevant objectives and outcomes.

	LESSON PLAN **Unit-VI**	2015-16
		Regulation: R12

Name of the Faculty: HIMAGIRI DANAPANA

Subject COMPILER DESIGN Subject Code 6756031

Unit **VI**

INSTRUCTIONAL OBJECTIVES:

Session No	**Topics to be covered**	**Time**	**Ref**	**Teaching Method**
1	Consideration for Optimization	**50 min**	TB1-8,9	Chalk& board Exercises
2	Scope of Optimization	**50 min**	TB1-8.4	Chalk& board Exercises
3	local optimization, loop optimization	**50 min**	TB-1 9.7	Chalk& board Exercises
4	frequency reduction	**50 min**	TB-1 9.7	Chalk& board Exercises
5	Folding	**50 min**	TB1-6.1	Chalk& board Exercises

On completion of this lesson the student shall be able to (Outcomes)

1.

2.

3.

4

	ASSIGNMENT **Unit-VI**	2015-16
		Regulation: R12

Assignment / Questions

1. Explain different principal sources of optimization technique with suitable examples.

2.Apply loop optimization and code motion for the following program code.

```
begin
for i := 1 to n do
for j:=1 to n do
c[i, j] :=0;
for i := 1 to n do
for j:=1 to n do
1 of 2
for k :=1 to n do
c[i, j] :=c[i , j] +a[i ,k] *b [k ,j]
end
```

3. (a) What is DAG? Construct the DAG for the following basic block

```
D := B_C
E :=A+B
B := B+C
A := E-D
```

(b) What are the legal evaluation orders and names for the values at the nodes for the DAG of problem (a).
i. Assuming A, B and C are alive at the end of the basic block?
ii. Assuming only A is live at the end?

4. (a) What is loop invariant operation? Write an algorithm for detecting loop in-variant computations.
(b) Consider the following program fragment code:

```
int p=1, k=0, n;
while(k<n)
{
p=2*p;
k=k+1;
```

} Apply the invariant operation elimination on the above program segment

5.Explain machine dependent code optimization with example

6.Write the importance of global code optimization. Explain redundant sub expression elimination technique across different blocks with example

7.Explain with example the various techniques in loop optimization

Signature of Faculty

Note: Mention for each question the relevant objectives and outcomes.

	LESSON PLAN **Unit-VII**	2015-16
		Regulation: R12

Name of the Faculty: HIMAGIRI DANAPANA

Subject COMPILER DESIGN Subject Code 6756031

Unit **VII**

INSTRUCTIONAL OBJECTIVES:

Session No	**Topics to be covered**	**Time**	**Ref**	**Teaching Method**
1	Flow graph	**50 min**	TB-1 9.1	Chalk& board Exercises
2	Data flow equation	**50 min**	TB1-9.2,RB4-17	Chalk& board Exercises
3	Global optimization	**50 min**	TB1-9.2,RB4-17	Chalk& board Exercises
4	Redundant sub expression elimination,	**50 min**	TB1-9.2,RB4-17	Chalk& board Exercises
5	Induction variable elements	**50 min**	RB4-18, TB1-9.1	Chalk& board Exercises
6	Live variable analysis	**50 min**	RB4-17 ,TB1-9.2	Chalk& board Exercises
7	Copy propagation	**50 min**	TB-2 6.7	Chalk& board Exercises

On completion of this lesson the student shall be able to

1.

2.

3.

4

	ASSIGNMENT **Unit-VII**	2015-16 Regulation: R12

Assignment / Questions

1. Describe, how redundant expression elimination can be done in loop optimization technique, during global optimization.
2. Consider the following program which counts the primes form 2 to n using the sieve method on a suitably large array

```
begin
read n
for i : = 2 to n do
a[i] : = true / * initialize */
count: = 0;
for i : 2 to n ** .5 do
if a [i] then /* i is a prime */
begin
count := count +1
for j : = 2 * i to n by i do
a[j] : =false
/ * j is divisible by i */
end ;
print count
end
```

(a) Propagate out copy statements wherever possible.
(b) Is loop jumping possible? If so, do it.
(c) Eliminate the induction variables wherever possible

3. (a) What is an Induction variable? Explain with an example.
(b) Discuss how induction variables can be detected and how transformation can be applied.

4. Describe, how redundant expression elimination can be done in loop optimization technique, during global optimization.

5. Explain about global data flow analysis. List data flow equations for reaching definition blocks and apply it on the above derived three-address code

Signature of Faculty

Note: Mention for each question the relevant objectives and outcomes.

	LESSON PLAN **Unit-VIII**	2015-16
		Regulation: R12

Name of the Faculty: HIMAGIRI DANAPANA

Subject COMPILER DESIGN Subject Code 6756031

Unit **VIII**

INSTRUCTIONAL OBJECTIVES:

Session No	**Topics to be covered**	**Time**	**Ref**	**Teaching Method**
1	Object code forms	**50 min**	TB-2 9.1	Chalk& board Exercises
2	Machine dependent code optimization	**50 min**	TB-2 9.2	Chalk& board Exercises
3	register allocation and assignment generic code generation algorithms	**50 min**	TB-1 8.6.2	Chalk& board Exercises
4	DAG for register allocation.	**50 min**	TB-1 8.5.1	Chalk& board Exercises
5	Object code forms	**50 min**	TB-2 9.1	Chalk& board Exercises

On completion of this lesson the student shall be able to

1.

2.

3.

4

	ASSIGNMENT **Unit-VIII**	2015-16
		Regulation: R12

Assignment / Questions

1.a) Explain the concept of object code forms.
(b) Generate optimal machine code for the following C program. [6+10]

```
main()
{
int i, a[10];
while (i<=10) a[i] =0
}
```

2. (a) Write an algorithm for generating code from a labeled tree.
(b) Construct a DAG for the following program code:

```
x=y*z
w=p+y
y=y*z
p=w-x .
```

3. (a) Explain the different issues in the design of a code generator.
(b) Generate code for the following C statements:

i. x= f(a) + f(a) + f(a)
ii. x= f(a) /g(b,c)
iii. x= f(f(a))
iv. x= ++f(a)

4.(a) Write about global register allocation strategy for loops.
(b) Explain code generation from DAG. For the following instructions construct DAG.

```
t1:=a / b
t2:=a/b
t3:=e - t2
t4:=t1 -t3
t5:=e+t2
t6:=t4 * t5.
}
```

5.(a) What are the various problems that arise in generating good code? Briefly explain them
(b) "Good Code generation requires an intimate knowledge of target machine".
Justify the statement with an appropriate example

Signature of Faculty

Note: Mention for each question the relevant objectives and outcomes.