



## Lecture Plan 2

Semester:-VII

Class :-ECS

Course Code:- :- EC-712-F

Subject:-Compiler Designing

Unit:-I

S. No.	Topic :-Structure of Compiler, Passes	Time Allotted:-
1.	Introduction Compiler is v complex in structure so we dVIIide it into a no. of passes ,each pass performs its own functions we discuss these passes here. .	<u>5min</u>
2	Division of the Topic  . <b>phase</b> -Lexical,syntax,intermediate code generation, code optimization and code generation  <b>Passes-definition</b> , method to reduce the no. of passes(back patching)	20 10
3.	Conclusion  Discussed general structure of compiler.	10
4	Question / Answer  Difference bet. Phases and passes.	5

Assignment to be given:-

nil

Reference Readings:-

Principles of Compiler Design by Alfred V.Aho & Jeffrey D.Ullman.

## Lecture Plan 3

Semester:-VII

Class :-ECS

Course Code:- :- EC-712-F

Subject:-Compiler Designing

Unit:-I

S. No.	Topic :-Lexical Analysis, Syntax Analysis, Intermediate code generator, Optimization.	Time Allotted:-
1.	Introduction Revision of last class.	<u>5</u>
2	Division of the Topic  Lexical Analysis-tokens(definition, finding the no. of tokens in an expression) Syntax Analysis-checking patterns of tokens, checking syntax of operators. Intermediate code generation-three address code, parse tree, two address code Optimization-local optimization, loop optimization	10 10 10 10
3.	Conclusion  Discussed about each phase of compiler with e.g.	5
4	Question / Answer  Find no. of tokens in statement While $A > B$ & $A \leq 2 * B - 5$ do $A = A + B$	5

Assignment to be given:-

Explain the phases of compiler with block diagram.

Reference Readings:-

Principles of Compiler Design by Alfred V.Aho & Jeffrey D.Ullman

## Lecture Plan 4

Semester:-VII

Class :-ECS

Course Code:- :- EC-712-F

Subject:-Compiler Designing

Unit:-I

S. No.	Topic :-Code Generation, Bookkeeping, Compiler,Writing tools, Bootstrapping.	Time Allotted:-
1.	Introduction  Revision of various phases of compiler and detailed study of code generation, bookkeeping and error handler phase	5
2	Division of the Topic 1) Code generation Conversion from high level code to assembly code m/c 2) Bookkeeping-how compiler store the record of tokens 3)Error handler –discussion about the possible error in each phase 4)Compiler writing tool-Compiler- compiler, Compiler generator, Translator writing system  5)Boot Strapping.	10 10 10 10
3.	Conclusion  All phases of compiler were discussed in details, using boot strapping method or cross compiling we can easily generate new compilers.	5
4	Question / Answer  Tell one error for each phase.	5

Assignment to be given:-

What is cross compiling and bootstrapping and difference between them.

Reference Readings:-

Principles of Compiler Design by Alfred V.Aho & Jeffrey D.Ullman

## Lecture Plan 5

Semester:-VII

Class :-ECS

Course Code:- :- EC-712-F

Subject:-Compiler Designing

Unit:-II

S. No.	Topic:-Role of Lexical Analyser, Design of Lexical Analysis.	Time Allotted:-
1.	Introduction  Lexical analyzer is the first phase of compiler it takes the source prog as input and produce tokens in output.	5
2	Division of the Topic  Need of lexical analysis Input buffering Method for designing lexical analyzer and implementation of lexical analyzer.	10 10 10
3.	Conclusion  After this lecture we can find the no. of tokens in prog..and we have discussed about the parse tree form by using these tokens.	5
4	Question / Answer  What are token and different types of tokens?	10

Assignment to be given:-

nil

Reference Readings:-

Principles of Compiler Design by Alfred V.Aho & Jeffrey D.Ullman

## Lecture Plan 6

Semester:-VII

Class :-ECS

Course Code:- :- EC-712-F

Subject:-Compiler Designing

Unit:-II

S. No.	Topic :-Regular Exp., NFA, DFA	Time Allotted:-
1.	Introduction Regular expressions are used as a source prog. For compiler writing tools like LEX, YACC	5
2	Division of the Topic  Strings and languages Regular expressions-definition Finite Automata-NFA,DFA	10 10 10
3.	Conclusion  Finite Automata are mathematical models of a m/c and we use reg. expressions as input for these m/c. These are used in first phase of compiler	5
4	Question / Answer  Basic characteristics of NFA and DFA.	

Assignment to be given:-

nil

Reference Readings:-

Principles of Compiler Design by Alfred V.Aho & Jeffrey D.Ullman

## Lecture Plan 7

Semester:-VII

Class :-ECS

Course Code:- :- EC-712-F

Subject:-Compiler Designing

Unit:-II

S. No.	Topic :-Conversion from Regular Exp to NFA, NFA to DFA, Minimization of DFA.	Time Allotted:-
1.	Introduction  We can easily convert a regular expression into NFA and vice versa These are helpful for selecting the source lang. of compiler.	5
2	Division of the Topic  1) Rules for converting Regular expression to NFA 2) Method for converting NFA to DFA 3) Methods for minimizing the no. of states.	10 10 10
3.	Conclusion  All topics were covered with examples.	5
4	Question / Answer  Make NFA from given regular expressions  (0+1)*011(011)* (011+110+0+1)*	10

Assignment to be given:-

110\*(110+000)\*

Reference Readings:-

Principles of Compiler Design by Alfred V.Aho & Jeffrey D.Ullman

## Lecture Plan 8

Semester:-VII

Class :-ECS

Course Code:- :- EC-712-F

Subject:-Compiler Designing

Unit:-II

S. No.	Topic :-LEX, Implementation of Lexical Analyzer.	Time Allotted:-
1.	Introduction  Lex is a tool which is used to make lexical analyzers it takes a source prog. In form of reg. expressions as input and produce a lexical analyzer phase as output.	5
2	Division of the Topic  Lex –auxiliary definitions Translation rules Implementation of lexical analyzer.	10 10 20
3.	Conclusion  Using Lex we generate lexical analyzer.	5
4	Question / Answer  nil	

Assignment to be given:-

nil

Reference Readings:-

Principles of Compiler Design by Alfred V.Aho & Jeffrey D.Ullman

## Lecture Plan 9

Semester:-VII

Class :-ECS

Course Code:- :- EC-712-F

Subject:-Compiler Designing

Unit:-III

S. No.	Topic:-Introduction to Grammar, Type of Grammar, Context free Grammar.	Time Allotted:-
1.	Introduction  A grammar involves four quan.- terminals, on terminals, start symbols and productions.	5
2	Division of the Topic  Definition of grammar Context free grammar Left and right derivations and parse trees	10 10 10
3.	Conclusion  All topics were covered successfully.	5
4	Question / Answer For the given reg. exp. $(a/b)(a/b/0/1)^*$ Generate a context free grammar.	5

Assignment to be given:-

nil

Reference Readings:-

Principles of Compiler Design by Alfred V.Aho & Jeffrey D.Ullman

## Lecture Plan 10

Semester:-VII

Class :-ECS

Course Code:- :- EC-712-F

Subject:-Compiler Designing

Unit:-I

S. No.	Topic:-Parse Tree, Ambiguity.	Time Allotted:-
1.	Introduction Parse tree is the pictorial representation for a given string, derived from a grammar and if a grammar generates more than one parse tree for a single string then it is called ambiguous grammar.	20
2	Division of the Topic  Ambiguous grammar, methods for remove ambiguity of grammar Examples.	20
3.	Conclusion  We can remove the ambiguity of the grammar so that there is a single step for the syntax analyzer phase to proceed in checking the syntax of the grammar.	5
4	Question / Answer  Stat → if cond then stat  if cond then stat else stat  other stat.	5

Assignment to be given:-

nil

Reference Readings:-

Principles of Compiler Design by Alfred V.Aho & Jeffrey D.Ullman

## Lecture Plan 11

Semester:-VII

Class :-ECS

Course Code:- :- EC-712-F

Subject:-Compiler Designing

Unit:-I

S. No.	Topic :-Capabilities of context free grammar Regular Exp. Vs Context free grammar	Time Allotted:-
1.	Introduction  There is method for converting a reg exp. Into context free grammar.	5
2	Division of the Topic  Reg exp verses context free grammar Egs of context free grammar Non context free Lang constructs	10 10 10
3.	Conclusion  All topics were covered with egs.	5
4	Question / Answer  Check whether lang $L=\{wcw w \text{ is in } (a/b)^*\}$ is context free lang or not.	10

Assignment to be given:-

nil

Reference Readings:-

Principles of Compiler Design by Alfred V.Aho & Jeffrey D.Ullman

## Lecture Plan 12

Semester:-VII

Class :-ECS

Course Code:- :- EC-712-F

Subject:-Compiler Designing

Unit:-I

S. No.	Topic :-Introduction to parser, Representation of parse tree	Time Allotted:-
1.	Introduction  Parser or syntax analyzer is the second phase of compiler it takes tokens as input and produce parse tree. as o/p	5
2	Division of the Topic  Introduction to the parser Type of parser Representation of parse tree.	10 10 10
3.	Conclusion  Parse tree was discussed in class and problems related to it were solved.	5
4	Question / Answer  Consider the grammar- $S \rightarrow iCtS$ $S \rightarrow iCtSeS$ $S \rightarrow a$ $C \rightarrow b$ Generate parse tree for $iw=ibtibtaea$	10

Assignment to be given:-  
nil

Reference Readings:-

Principles of Compiler Design by Alfred V.Aho & Jeffrey D.Ullman

## Lecture Plan 13

Faculty:-  
Code:- CSE-403-C

Semester:-VII

Class:-CSE I,CSEII

Course

Subject:-Compiler Designing

Unit:-VII

S. No.	Topic :-Shift reduce Parser, Handle, Handle pruning.	Time Allotted:-
1.	Introduction  Shift reduce parser uses bottom up approach means we start from the word and generate parse tree in reverse. If we reach up to start symbol then the word is recognized otherwise not	5
2	Division of the Topic  Shift reduce parser Handle Handle pruning.	10 10 10
3.	Conclusion  Handles are used to reduce the given words upto its start symbol ,first we find handles and then we replace them from the production rules used in grammar.	5
4	Question / Answer  What is handle, handle pruning.	10

Assignment to be given:-

Nil.

Reference Readings:-

Principles of Compiler Design by Alfred V.Aho & Jeffrey D.Ullman

## Lecture Plan 14

Faculty:-  
Code:- CSE-403-C

Semester:-VII

Class:-CSE I,CSEII

Course

Subject:-Compiler Designing

Unit:-VII

S. No.	Topic :-Stack Implementation of shift Reduce parser.	Time Allotted:-
1.	Introduction In last lecture we discussed about shift reduce parser here we study the method how to implement a shift reduce parser	10
2	Division of the Topic  Implementation of shift reduce parser-There are four actions 1)shift 2) Reduce 3) Accept 4) Error.	10  10
3.	Conclusion  Shift reduce parser is completed	5
4	Question / Answer Consider the grammar- $E \rightarrow E+E$ $E \rightarrow E * E$ $E \rightarrow (E)$ $E \rightarrow id$  Make a shift reduce parser for the i/p string $id +id * id$	15

Assignment to be given:-

nil

Reference Readings:-

Principles of Compiler Design by Alfred V.Aho & Jeffrey D.Ullman

**Lecture Plan 15**Semester:-VIIClass :-ECSCourse Code:- :- EC-712-FSubject:-Compiler DesigningUnit:-VII

S. No.	Topic:-Operator Precedence Parsing.	Time Allotted:-
1.	Introduction  There are some grammars that don't have productions in which there E on right hand side or has two adjacent no terminals these grammars are called operator grammars and we generates a parser for these by using operator precedence parsing method.	5
2	Division of the Topic  1)Operator grammar 2)Operator precedence relations 3)operator precedence relations from associativity and precedence 4)operator precedence parsing algo.	10 10 10 10
3.	Conclusion  Operator precedence parser is used to generate parse tree for operator grammar.	5
4	Question / Answer  What is leading(A) and trailing(A)	

Assignment to be given:-

nil

Reference Readings:-

Principles of Compiler Design by Alfred V.Aho &amp; Jeffrey D.Ullman

## Lecture Plan 16

Semester:-VII

Class :-ECS

Course Code:- :- EC-712-F

Subject:-Compiler Designing

Unit:-VII

S. No.	Topic:-.Top down parsing	Time Allotted:-
1.	Introduction Top down parsing starts from the starting symbol and find the word after applying a no. of productions. .	5
2	Division of the Topic  1) Backtracking. 2) Left recursion, Elimination of left recursion. 3)Recursive descent parsing 4) Left factoring.	10 10 10 10
3.	Conclusion  Top down parser are good but there is problem of backtracking	
4	Question / Answer  Nil	

Assignment to be given:-

nil

Reference Readings:-

Principles of Compiler Design by Alfred V.Aho & Jeffrey D.Ullman

## Lecture Plan 17

Semester:-VII

Class :-ECS

Course Code:- :- EC-712-F

Subject:-Compiler Designing

Unit:-VII

S. No.	Topic:-.Predictive parser.	Time Allotted:-
1.	Introduction PredictVe parser is helpful to implement recursive descent parsing it is type of top down parsing. .	5
2	Division of the Topic  Predictive parser-first and follow Method for constructing predictive parsing table.	20 20
3.	Conclusion  All the parsers are discussed with numerical egs.	5
4	Question / Answer  Nil	

Assignment to be given:-

nil

Reference Readings:-

Principles of Compiler Design by Alfred V.Aho & Jeffrey D.Ullman

## Lecture Plan 18

Semester:-VII

Class :-ECS

Course Code:- :- EC-712-F

Subject:-Compiler Designing

Unit:-VII

S. No.	Topic :-LR parsers.	Time Allotted:-
1.	<p>Introduction</p> <p>These parsers scans from left to right and constructs a rightmost derivation in reverse</p> <p>It consists of two parts a driver routine and a parsing table. These are of three types. Drivng routine is same for all while parsing table is different.</p>	5
2	<p>Division of the Topic</p> <p>LR parser-Definition</p> <p>Working-1.shift 2 Reduce 3 Accept 4 Error.</p> <p>Eg.</p>	20 20
3.	<p>Conclusion</p> <p>Out of three types SLR parsers are simple but having v. large table size .Canonical LR parsers are the best.</p> <p>We will discuss all these in next lectures.</p>	5
4	<p>Question / Answer</p> <p>Nil</p>	

Assignment to be given:-

Reference Readings:- Principles of Compiler Design by Alfred V.Aho & Jeffrey D.Ullman

## Lecture Plan 19

Semester:-VII

Class :-ECS

Course Code:- :- EC-712-F

Subject:-Compiler Designing

Unit:-VII

S. No.	Topic:-SLR parsers.	Time Allotted:-
1.	<p>Introduction</p> <p>These parsers scans from left to right and constructs a rightmost derivation in reverse</p> <p>It consists of two parts a driver routine and a parsing table</p>	5
2	<p>Division of the Topic</p> <ul style="list-style-type: none"><li>1) LR (0) Items –Closure,GOTO,Set of item construction</li><li>2) Method for constructing SLR parsing table.</li><li>3) Example</li></ul>	20 15 5
3.	<p>Conclusion</p> <p>SLR was taught successfully.</p>	5
4	<p>Question / Answer</p> <p>nil</p>	

Assignment to be given:-nil

Reference Readings:- Principles of Compiler Design by Alfred V.Aho & Jeffrey D.Ullman

## Lecture Plan 20

Semester:-VII

Class :-ECS

Course Code:- :- EC-712-F

Subject:-Compiler Designing

Unit:-VII

S. No.	Topic :-CANONICAL LR,LALR	Time Allotted:-
1.	Introduction  Canonical LR parser is most powerful among all three parsers. To max space utilization we convert canonical LR parser to LALR parser but it is time consuming.	5
2	Division of the Topic  Method for constructing canonical LR parsing table. 1)Closure(I) 2)GOTO(I,X) 3) Method for table. Construction. Method for constructing LALR parsing table	10 10 10 10
3.	Conclusion  All topics are covered in details.	5
4	Question / Answer nil	

Assignment to be gVIIen:-

Reference Readings:- Principles of Compiler Design by Alfred V.Aho & Jeffrey D.Ullman

## Lecture Plan 21

Semester:-VII

Class :-ECS

Course Code:- :- EC-712-F

Subject:-Compiler Designing

Unit:-V

S. No.	Topic:-Syntax directed translation scheme.	Time Allotted:-
1.	Introduction We will discuss here different types of intermediate codes generation methods. Syntax directed translation scheme allow some semantic actions are to be attached with context free grammar. If any production rules match by syntax analyzer phase then the action attached with that production takes place	5
2	Division of the Topic  1) Syntax directed translation schemes-Semantic actions, translation on parse tree.  2)Implementation of syntax directed translation	20  20
3.	Conclusion  Intermediate codes are helpful in making the programs easier to understand Like we can convert high level Lang into assembly codes which are easier to understand by translator	5
4	Question / Answer nil	

Assignment to be given:-nil

Reference Readings:- Principles of Compiler Design by Alfred V.Aho & Jeffrey D.Ullman

Doc. No.: DCE/0/15

Revision :00

## Lecture Plan 22

Semester:-VII

Class :-ECS

Course Code:- :- EC-712-F

Subject:-Compiler Designing

Unit:-VII

S. No.	Topic: Three Address Codes, Quadruples, Triples.	Time Allotted:-
1.	Introduction  These are types of intermediate codes .In three address codes each statement is converted into an equivalent statement having three addresses.	5
2	Division of the Topic  Three address codes Quadruples Triples Indirect triples.	10 10 10 10
3.	Conclusion  Using these methods each statement of high level Lang is converted into intermediate codes.	5
4	Question / Answer nil	

Assignment to be given:-nil

Reference Readings:- Principles of Compiler Design by Alfred V.Aho & Jeffrey D.Ullman

**Lecture Plan 23**Semester:-VIIClass :-ECS \_\_\_\_\_ Course Code:- :- EC-712-FSubject:-Compiler DesigningUnit:-VI

<b>S. No.</b>	<b>Topic :-Content of symbol table</b>	<b>Time Allotted:-</b>
1.	<p>Introduction</p> <p>Symbol Tables are used to store different tokens and special symbols with their attributes. Symbol tables are used by each phase of compiler every phase can take input from symbol tables and store its output in it</p>	5
2	<p>Division of the Topic</p> <p>1) Contents of symbol table 2) Names and symbol table records 3) Reusing symbol table records</p>	15 15 10
3.	<p>Conclusion</p> <p>Symbol tables can be stored in a no. of ways. We generally stores token names there types and there values .There should be easy methods to store and retrieve records from symbol tables.</p>	2
4	<p>Question / Answer</p> <p>What is symbol table?</p>	3

Assignment to be given:-nilReference Readings:- Principles of Compiler Design by Alfred V.Aho & Jeffrey D.Ullman

## Lecture Plan 24

Semester:-VII

Class :-ECS

Course Code:- :- EC-712-F

Subject:-Compiler Designing

Unit:-VII

S. No.	Topic:-Data Structure for symbol tables.	Time Allotted:-
1.	Introduction  We can implement symbol tables by using diff. data structures. There are a no. of methods are given below.	5
2	Division of the Topic  1. List 2. Self organizing list. 3. Search trees 4. Hash table 5. Representation scope information.	5 5 5 5 10
3.	Conclusion Each method has its own adv. And disadvantage. We use methods according to our size of data but search tree and hash tables' given best. Results in terms of time.]	5
4	Question / Answer  What are search trees and hash tables.	10

Assignment to be given:-nil

Reference Readings:- Principles of Compiler Design by Alfred V.Aho & Jeffrey D.Ullman

## Lecture Plan 25

Semester:-VII

Class :-ECS

Course Code:- :- EC-712-F

Subject:-Compiler Designing

Unit:-VII

S. No.	Topic:-Error detection and recovery.	Time Allotted:-
1.	Introduction  Prog. Submitted to compiler may contain a no. of errors and there could be a no. of errors generated during various phases .To handle these types of errors Error handler phase is helpful.	5
2	Division of the Topic Error-Definition, Reporting errors ,source of errors Syntactic errors Semantic errors Dynamic errors	
3.	Conclusion Syntactic error comes because of there syntax is wrong. Semantic errors are due to type checking. Dynamic errors are generated during compiling and execution phase.	
4	Question / Answer  Ex`plain various types of errors.	

Assignment to be given:-

Reference Readings:- Principles of Compiler Design by Alfred V.Aho & Jeffrey D.Ullman

## Lecture Plan 26

Semester:-VII

Class :-ECS

Course Code:- :- EC-712-F

Subject:-Compiler Designing

Unit:-VII

S. No.	Topic :-Lexical Phase error, Syntactic phase error.	Time Allotted:-
1.	Introduction  The type of token error generated during lexical phase are covered here and there are methods to correct them	5
2	Division of the Topic  Lexical errors Minimum distance matching Syntactic phase error –Time of detection,panic mode ,error recovery in operator precedence parsing.	20  20
3.	Conclusion  We have discussed about different types of errors and how to remove them at prog level.	2
4	Question / Answer  What is distance matching ,in which phase this method used.	3

Assignment to be given:-nil

Reference Readings:- Principles of Compiler Design by Alfred V.Aho & Jeffrey D.Ullman

## Lecture Plan 27

Semester:-VII

Class :-ECS

Course Code:- :- EC-712-F

Subject:-Compiler Designing

Unit:-VII

S. No.	Topic :-Syntactic phase error.	Time Allotted:-
1.	Introduction  If proper syntax is not given then we encounter syntactic error.  Context free grammar is used in syntax phase are helpful here.	5
2	Division of the Topic  error recovery in operator precedence parsing continue Error recovery in LR parsers. Recursive descent parser	10 20 5
3.	Conclusion  In this way many errors can be removed by various phases of compiler	10
4	Question / Answer  nil	

Assignment to be given:-

nil

Reference Readings:- Principles of Compiler Design by Alfred V.Aho & Jeffrey D.Ullman

## Lecture Plan 28

Semester:-VII

Class :-ECS

Course Code:- :- EC-712-F

Subject:-Compiler Designing

Unit:-VII

S. No.	Topic:-Code Generation	Time Allotted:-
1.	Introduction  This is the final phase of compiler and it produces either assembly codes or m/c codes.	5
2	Division of the Topic  Object codes Environment of code generator Problems in code generation.	10 10 10
3.	Conclusion  There are diff types of object codes like m/c code, assembly code or the codes produced from intermediate lang .Different type of environments are required for different type of codes.,.	5
4	Question / Answer  Difference bet. Assembly codes and intermediate codes.	10

Assignment to be given:- nil

Reference Readings:- Principles of Compiler Design by Alfred V.Aho & Jeffrey D.Ullman

## Lecture Plan 29

Semester:-VII

Class :-ECS

Course Code:- :- EC-712-F

Subject:-Compiler Designing

Unit:-VII

S. No.	Topic :-Machine dependent model	Time Allotted:-
1.	<p>Introduction</p> <p>Good code generator requires an intimate knowledge of target m/c.Here we discuss about a m/c –PDP11 and try to find out which types of problems could be faced in code generation</p>	5
2	<p>Division of the Topic</p> <p>Example for different addressing modes require different types of code generator., Simple code generator-Introduction next use information register description address description code generation algo.</p>	10 10 10 10
3.	<p>Conclusion</p> <p>We have discussed about various problems faced in code generation by using a simple eg. For code generation .</p>	5
4	<p>Question / Answer</p> <p>nil</p>	

Assignment to be given:-

nil

Reference Readings:- Principles of Compiler Design by Alfred V.Aho & Jeffrey D.Ullman

## Lecture Plan 30

Semester:-VII

Class :-ECS

Course Code:- :- EC-712-F

Subject:-Compiler Designing

Unit:-VII

S. No.	Topic :- <b>Optimization phase</b>	Time Allotted:-
1.	Introduction Optimization of codes is done to minimize the space and to maximize the speed of prog. Execution. Some criteria are there like optimization preserved meaning of source prog>no optimization should map a correct prog> into an incorrect prog and effort applied for optimization should be reasonable	5
2	Division of the Topic  Principal source of optimization are 1)Inner loops  2)Lang. implementation details inaccessible to users 3)Optimization in sub expressions 4) Algo. optimization	10 10 10 10
3.	Conclusion Optimization can be done at various points the best one is if we do it at algorithm level .Loop optimization can be done easily by compilation and optimization phase.	2
4	Question / Answer  How you will optimize a loop. Also give an eg. For optimization in expressions.	3

Assignment to be given:-

nil

Reference Readings:- Principles of Compiler Design by Alfred V.Aho & Jeffrey D.Ullman

Doc. No.: DCE/0/03  
Revision :00

## Lecture Plan 31

Semester:-VII

Class :-ECS

Course Code:- :- EC-712-F

Subject:-Compiler Designing

Unit:-VII

S. No.	Topic:-Register allocation for temp and user defined variable.	Time Allotted:-
1.	Introduction  Here we discuss various strategies for deciding what names in a prog. Should reside in registers, a problem often referred to as register allocation, and in which registers each should reside (register assignment.)	5
2	Division of the Topic  Introduction Global register allocation Register assignment for outer loops	10 10 10
3.	Conclusion  For register allocations we divide our prog. Into different basic groups after that we assess each group one by one so that multiple declaration of same variable should not be there.	5
4	Question / Answer  What is basic block. How you will create no. of basic blocks. in a prog.	10

Assignment to be given:-

nil

Reference Readings:- Principles of Compiler Design by Alfred V.Aho & Jeffrey D.Ullman