

# FR. Conceicao Rodrigues College Of Engineering

Father Agnel Ashram, Bandstand, Bandra-west, Mumbai-50

**Department of Computer Engineering**

**B.E. (Computer) (semester VI)**

**(2018-2019)**

## **Course Outcomes & Assessment Plan**

**Subject:** System Programming Compiler Construction (CPC601)

**Credits-5**

### **Syllabus:**

#### **1. Introduction to System Software**

Concept of System Software, Goals of system softwares, system program and system programming, Introduction to various system programs such as Assembler, Macro processor, Loader, Linker, Compiler, Interpreter, Device Drivers, Operating system, Editors, Debuggers

#### **2. Assemblers**

Elements of Assembly Language programming, Assembly scheme, pass structure of assembler, Assembler Design: Two pass assembler Design and single pass Assembler Design for Hypothetical / X86 family processor, data structures used.

#### **3. Macros & Macro processors**

Introduction, Macro definition and call, Features of Macro facility: Simple, parameterized, conditional and nested. Design of single pass macro processor, data structures used

#### **4. Loaders and Linkers**

Introduction, functions of loaders, Relocation and Linking concept, Different loading schemes: Relocating loader, Direct Linking Loader, Dynamic linking and loading.

#### **Compilers: Analysis Phase**

Introduction to compilers, Phases of compilers: Lexical Analysis- Role of Finite State Automata in Lexical Analysis, Design of Lexical analyser, data structures used . Syntax Analysis- Role of Context Free Grammar in Syntax analysis, Types of Parsers: Top down parser- LL(1), Bottom up parser- Operator precedence parser, SLR Semantic Analysis, Syntax directed definitions.

#### **Compilers: Synthesis Phase**

Intermediate Code Generation: Types of Intermediate codes: Syntax tree, Postfix notation, Three address codes: Triples and Quadruples. Code Optimization: Need and sources of optimization, Code optimization techniques: Machine Dependent and Machine Independent. Code Generation: Issues in the design of code generator, code generation algorithm. Basic block and flow graph.

**Reference Books:** 1. Modern Compiler. Implementation in Java, Second. Edition. Andrew W. Appel Princeton University. Jens Palsberg Purdue University. CAMBRIDGE. 2. Crafting a compiler with C, Charles N. Fischer, Ron K. Cytron, Richard J. LeBlanc

#### **Online resources:**

NPTEL Links uploaded on moodle

#### **Course objectives:**

1. To understand the role and functioning of various system programs over application program.
2. To understand basic concepts and designing of assembler, Macro processor and role of static and dynamic loaders and linkers.

3. To understand the need to follow the syntax in writing an application program and to learn the how the analysis phase of compiler is designed to understand the programmer's requirements without ambiguity.
4. To synthesize the analysis phase outcomes to produce the object code that is efficient in terms of space and execution time.

**Course Outcomes:**

| <b>Course Outcome</b> | <b>After successful completion of the course students should be able to</b>  |
|-----------------------|--|
| CO1                   | Identify the different system programs and their utility.( B2-Understanding) |
| CO2                   | Design basic two pass Assembler and Macro Pre-processor (B2-Understanding)   |
| CO3                   | Acquire knowledge on various phases of Compiler. (B2-Understanding)          |
| CO4                   | Design various parsing techniques (B2-Understanding)                         |

Relationship of course outcomes with program outcomes: Indicate 1 (low importance), 2 (Moderate Importance) or 3 (High Importance) in respective mapping cell.

|              | PO1<br>(Engg Know) | PO2<br>(Ana) | PO3<br>(Design) | PO4<br>(investiga) | PO5<br>(tools) | PO6<br>(eng Soc) | PO7<br>(Env) | PO8<br>(Eth) | PO9<br>(ind Team) | PO10<br>(comm.) | PO11<br>(PM) | PO12<br>(life Long) |
|--------------|--------------------|--------------|-----------------|--------------------|----------------|------------------|--------------|--------------|-------------------|-----------------|--------------|---------------------|
| CPC601.1     | 3                  | 1            |                 |                    |                |                  |              |              |                   |                 |              |                     |
| CPC601.2     | 3                  | 2            | 3               |                    |                |                  |              |              |                   |                 |              |                     |
| CPC601.3     | 3                  | 2            | 1               |                    | 1              |                  |              |              |                   |                 |              |                     |
| CPC601.4     | 3                  | 2            | 3               |                    | 1              |                  |              |              |                   |                 |              |                     |
| Course To PO |                    |              |                 |                    |                |                  |              |              |                   |                 |              |                     |

| CO       | PSO1 | PSO2 |
|----------|------|------|
| CPC601.1 | 2    |      |
| CPC601.2 | 2    |      |
| CPC601.3 | 2    |      |
| CPC601.4 | 2    |      |

|               |  |  |
|---------------|--|--|
| Course to PSO |  |  |
|---------------|--|--|

### Justification

**PO1:** This subject all COs are mapped to PO1 because engineering graduates will be able to apply the knowledge of System Programming to solve complex engineering problems.

**PO2:** This subject all COs are mapped to PO2 because the students analyze the given problem statement

**PO3:** This subject all COs are mapped to PO3 because the students design the various system soft wares.

**PO5:** CO601.3 and CO601.4 are mapped to PO5 because the students use the tools like Lex and Yacc to implement phases of compilers.

**PSO1:** All COs are mapped to PSO1 because the graduates will be able to apply knowledge of system program to simulate the real world problem.

### CO Assessment Tools:

**CPC601.1:** Direct Methods(80%): Test1, Quiz UniExamTh

$$CO1dm = 0.3T1 + 0.2assign + 0.1Lab + 0.2Q + 0.1UTh + 0.1UTPr$$

InDirect Methods(20%): Course exit survey

$$CO1idm$$

$$CSC302.1 = 0.8 * CO1dm + 0.2 * CO1idm$$

| Direct Methods      | Weightage | Target  | Date              | Marks                  |
|---------------------|-----------|---|-------------------|------------------------|
| Test 1&Test2        | 0.3       | 60% students will score minimum 60% marks (i.e. 3 or more out of 5)     | 05-02-19 & 9-4-19 | Q-1 (05M)<br>Q-2 (05M) |
| Assignment          | 0.2       | 60% students will score minimum 70% marks (i.e. 7 or more out of 10)    | 20-02-19          | 10M                    |
| Quiz1               | 0.2       | 60% students will score minimum 70% marks (i.e. 14 or more out of 20)   | 28-02-19          | 10M                    |
| Lab                 | 0.1       | 60% students will score minimum 70% marks (i.e. 7 or more out of 10)    | Expt 3 and 8      | 10M                    |
| Uni Theory exam     | 0.1       | 60% students will score minimum 65% marks (i.e. 52 or more out of 80)   | 17-05-19          | 80M                    |
| Uni. Practical Exam | 0.1       | 60% students will score minimum 70% marks (i.e. 17.5 or more out of 25) | 26-4-19 & 4-5-19  | 25M                    |

**CPC601.2:** Direct Methods(80): Test1,2 Quiz Lab Assignment UniExamTh

$$CO2dm = 0.2T1, T2 + 0.2Q + 0.4Lab + 0.1UTh + 0.1UTPr$$

InDirect Methods(20%): Course exit survey

$$CO2idm$$

$$CSC302.2 = 0.8 * CO2dm + 0.2 * CO2idm$$

| Direct Methods | Weightage | Target  | Date              | Marks                  |
|----------------|-----------|---|-------------------|------------------------|
| Test 1&Test2   | 0.2       | 60% students will score minimum 60% marks (i.e. 3 or more out of 5) | 05-02-19 & 9-4-19 | Q-1 (05M)<br>Q-2 (05M) |
| Quiz2          | 0.2       | 60% students will score minimum 70%                                 | 30-03-19          | 10M                    |

|                     |      |   |                       |     |
|---------------------|------|---|-----------------------|-----|
|                     |      | marks (i.6. 14 or more out of 20)                                       |                       |     |
| Lab                 | 0.3  | 60% students will score minimum 70% marks (i.6. 14 or more out of 20)   | EXPERIMENT NO 1,2 & 3 | 10M |
| Uni Theory exam     | 0.15 | 60% students will score minimum 65% marks (i.6. 52 or more out of 80)   | 17-05-19              | 80M |
| Uni. Practical Exam | 0.15 | 60% students will score minimum 70% marks (i.6. 17.5 or more out of 25) | 26-4-19 & 4-5-19      | 25M |

**CPC601.3: Direct Methods(80%):** Test1 Quiz Lab UniExamTh

$$CO3dm = 0.1T1 + 0.2Q++0.4Lab + 0.3UTh$$

**InDirect Methods(20%):** Course exit survey

$$CO3idm$$

$$CSC302.3 = 0.8*CO3dm + 0.2* CO3idm$$

| Direct Methods      | Weightage | Target  | Date                       | Marks     |
|---------------------|-----------|---|----------------------------|-----------|
| Test 1              | 0.1       | 60% students will score minimum 60% marks (i.6. 6 or more out of 10)    | 05-02-19                   | Q-3 (10M) |
| Quiz2               | 0.2       | 60% students will score minimum 70% marks (i.6. 14 or more out of 20)   | 02-04-19                   | 20M       |
| Lab                 | 0.4       | 60% students will score minimum 70% marks (i.6. 14 or more out of 20)   | Experiments 4, 5 , 6 and 7 | 10M       |
| Uni Theory exam     | 0.2       | 60% students will score minimum 65% marks (i.6. 52 or more out of 80)   | 17-05-19                   | 80M       |
| Uni. Practical Exam | 0.1       | 60% students will score minimum 70% marks (i.6. 17.5 or more out of 25) | 26-4-19 & 4-5-19           | 25M       |

**CPC601.4: Direct Methods(80%):** Test2 Lab Assignment UniExamTh

$$CO4dm = 0.2T2+0.3Lab +0.2A+ 0.3Uth$$

**InDirect Methods(20%):** Course exit survey

$$CO4idm$$

$$CSC302.4 = 0.8*CO4dm + 0.2* CO4idm$$

| Direct Methods | Weightage | Target  | Date                       | Marks     |
|----------------|-----------|---|----------------------------|-----------|
| Test 1         | 0.2       | 60% students will score minimum 60% marks (i.6. 6 or more out of 10)  | 05-02-19                   | Q-3 (10M) |
| Test2          | 0.2       | 60% students will score minimum 60% marks (i.6. 6 or more out of 10)  | 09-04-19                   | Q-3 (10M) |
| Assignment     | 0.1       | 60% students will score minimum 70% marks (i.6. 7 or more out of 10)  | 02-04-19                   | 20M       |
| Lab            | 0.2       | 60% students will score minimum 70% marks (i.6. 14 or more out of 20) | Experiments 4, 5 , 6 and 7 | 10M       |

|                     |     |   |                  |     |
|---------------------|-----|---|------------------|-----|
| Uni Theory exam     | 0.2 | 60% students will score minimum 65% marks (i.e. 52 or more out of 80)   | 17-05-19         | 80M |
| Uni. Practical Exam | 0.1 | 60% students will score minimum 70% marks (i.e. 17.5 or more out of 25) | 26-4-19 & 4-5-19 | 25M |

### **Content beyond syllabus:**

Case study on Natural Language processing – Students will understand to apply the knowledge of phases of compiler to NLP.

### **Course Outcomes Target:**

*Upon completion of this course students will be able to:*

**CPC601.1:** Identify the different system programs and their utility

**Target level: 2.5**

**CPC601.2:** Design basic two pass assembler and Macro Pre-processor

**Target level: 2.5**

**CPC601.4:** Acquire various phases of Compiler

**Target level: 2.5**

**CPC601.5:** Design various parsing techniques

**Target level: 2.5**

### **RUBRICS for the SPCC Lab Experiments**

|   |                       |                                 |   |  |   |
|---|-----------------------|---------------------------------|---|--|---|
| 1 | Time Line (2)         | N.A                             | Two sessions late (0)   | One session late (1)                                     | Early or on time (2)                        |
| 2 | Output (3)            | Practical not performed.<br>(0) | Practical performed but failed to show output due to some error.<br>(1) | Output shown but not as expected (Partial output)<br>(2) | Expected output shown<br>(3)                |
| 3 | Code optimization (3) | Practical not performed<br>(0)  | Code is unstructured and difficult to understand(1)                     | The code is structured<br>(2)                            | The code is structured and optimized<br>(3) |

|   |                                      |     |  |   |   |
|---|--------------------------------------|-----|--|---|---|
| 4 | Knowledge about the topic<br><br>(2) | N.A | Not able to answer any question<br><br>(0) | Able to answer few questions<br><br>(1) | Answered all the questions with relevant explanation<br><br>(2) |
|---|--------------------------------------|-----|--|---|---|

**Rubrics for the Assignment :**

| Indicator                                  | Very Poor                    | Poor   | Average   | Good   | Excellent  |
|--|------------------------------|--|---|--|--|
| <b>On time Submission (2)</b>              | Assignment not submitted (0) | More than two session late (0.5)                   | Two sessions late (1)   | One session late (1.5)   | Early or on time (2)   |
| <b>Organization (2)</b>                    | N/A                          | Very poor readability and not structured (0.5)     | Poor readability and somewhat structured (1)                          | Readable with one or two mistakes and structured (1.5)                       | Very well written and structured without any mistakes (2)      |
| <b>Level of content (4)</b>                | N/A                          | Major points are omitted / addressed minimally (1) | All major topics are covered, the information is accurate. (2)        | Most major and some minor criteria are included. Information is Accurate (3) | All major and minor criteria are covered and are accurate. (4) |
| <b>Depth and breadth of discussion (2)</b> | N/A                          | None in evidence; superficial at most (0.5)        | Minor points/information may be missing and discussion is minimal (1) | Discussion centers on some of the points and covers them adequately (1.5)    | Information is presented in depth and is accurate (2)          |

**FR. CONCEICAO RODRIGUES COLLEGE OF ENGG.  
Fr. Agnel Ashram, Bandstand, Bandra (W) Mumbai 400 050.**

**I UNIT TEST**

**SEMESTER: VI**

**BRANCH: COMPUTER**

**SUBJECT: System Programming Compiler Construction (CPC601)**

**MAX. MARKS: 20**

**DATE: 05/02/2019**

**TIME: 10AM -11AM**

**Course Outcomes:**

CPC601.1 Identify the different system programs and their utility.

CPC601.2 Design basic two pass Assembler and Macro Pre-processor.

CPC601.3 Acquire knowledge on various phases of Compiler.

CPC601.4 Design various parsing techniques.

| Q.No | Answer the following   | Marks | CO       |
|------|--|-------|----------|
| Q-1  | List and explain various system programs, Draw a neat labelled diagram showing their interconnection.<br><br><b>OR</b><br><br>What are System Softwares, Explain the Goals of system Softwares   | [5]   | CPC601.1 |
| Q-2  | Convert the Given Source Program into Intermediate code also populate all the required data structures.<br><b>START 300</b><br><b>MOVEM BREG , DATA1</b><br><b>LA1: PRINT DATA2</b><br><b>COMP DREG , ='43'</b><br><b>LTORG</b><br><b>DATA1 DS 49</b><br><b>LA2: EQU LA1</b><br><b>ORIGIN 550</b><br><b>SUB AREG , ='22'</b><br><b>LTORG</b><br><b>DATA2 DC 87</b><br><b>END</b> | [5]   | CPC601.2 |





**FR. CONCEICAO RODRIGUES COLLEGE OF ENGG.**  
**Fr. Agnel Ashram, Bandstand, Bandra (W) Mumbai 400 050.**

**II UNIT TEST**

**SEMESTER: VI**

**BRANCH: COMPUTER**

**SUBJECT: System Programming Compiler Construction (CPC601)**

**MAX. MARKS: 20**

**DATE: 09/04/2019**

**TIME: 10AM -11AM**

**Course Outcomes:**

CPC601.1 Identify the different system programs and their utility.

CPC601.2 Design basic two pass Assembler and Macro Pre-processor.

CPC601.4 Design various parsing techniques.

| Q.No | Answer the following   | Marks | CO                        |
|------|--|-------|---------------------------|
| Q-1  | Convert the Given Source Program into target code using two pass assembler also populate all the required data structures.<br><br><pre> START 300 MOVEM CREG , INFO1 LA1: PRINT DATA2 ADD AREG , ='32' LTORG INFO1 DS 44 LA2: EQU LA1 ORIGIN 550 SUB DREG , ='33' LTORG DATA2 DC 66 END </pre> | [5]   | CPC601.1                  |
| Q-2  | Explain the working of single pass Macro processor with neat flowcharts and databases (Clearly Show Entries in database)<br><br><pre> MACRO EVAL &amp;X, &amp;Y=, &amp;OP=ADD LOAD &amp;X &amp;OP &amp;Y STORE &amp;X MEND </pre>  | [5]   | CPC601.2                  |
| Q-3  | Design LR(1) and LALR parser for the grammar given below<br>$S \rightarrow CC$<br>$C \rightarrow cC$<br>$C \rightarrow d$  | [10]  | CPC601.3<br>&<br>CPC601.4 |

## Assignment-1

---

**Note: Date of submission: 25/3/19**

**CO4:** Design various parsing techniques.

1. Design operator precedence parser for the grammar

$E \rightarrow E+E \mid E * E \mid id$

2. Design predictive parser for the grammar given below

$E \rightarrow T \text{Edash}$

$\text{Edash} \rightarrow +T \text{Edash} \mid \text{NULL}$

$T \rightarrow (E) \mid id$

Check whether input **id\*id+** is valid

3. Design LR(0) parser table for the grammar

$E \rightarrow E+T \mid T$

$T \rightarrow id$

4. Design LR (1) and LALR parser for the grammar given below

$A \rightarrow (A)$

$A \rightarrow a$

### RUBRICS

1. Timeline (2)
2. Organization (2)
3. Steps of Design(6)

## **SPCC ASSIGNMENT NO 2**

**D0A: 22/2/19**

- 1. Explain various functions of Loaders. Compare linking loader and linkage loader.**
- 2. Discuss various intermediate code forms in detail**
- 3. Differentiate between :**
  - a. Compiler and Interpreter**
  - b. System software and Application software**

Father Agnel Ashram, Bandstand, Bandra-west, Mumbai-50  
**Department of Computer Engineering**  
**T.E. (Computer) (semester VI)**  
**(2018-2019)**

**Lesson Plan: SPCC**

**Semester VI**

**Year: 2018-19**

**Modes of Content Delivery:**

|     |                     |      |                                |     |                  |
|-----|---------------------|------|--------------------------------|-----|------------------|
| I   | Class Room Teaching | v    | Self Learning Online Resources | Ix  | Industry Visit   |
| ii  | Tutorial            | vi   | Slides                         | X   | Group Discussion |
| iii | Remedial Coaching   | vii  | Simulations/Demonstrations     | xi  | Seminar          |
| Iv  | Lab Experiment      | viii | Expert Lecture                 | xii | Case Study       |

| <b>Lect. No.</b> | <b>Portion to be covered</b>   | <b>Planned date</b> | <b>Actual date</b> | <b>Content Delivery Method/Learning Activities</b>                         |
|------------------|--|---------------------|--------------------|--|
| 1                | Introduction to compilers, Phases of compilers   | 02/01/2019          | 02/01/2019         | Class Room Teaching, Case Study, Lab Experiment Simulations/Demonstrations |
| 2                | Lexical Analysis- Role of Finite State Automata in Lexical Analysis  | 04/01/2019          | 04/01/2019         | Class Room Teaching, Case Study, Lab Experiment                            |
| 3                | Design of Lexical analyser, data structures used . Syntax Analysis- Role of Context Free Grammar in Syntax analysis, | 09/01/2019          | 07/01/2019         | Class Room Teaching, Case Study, Lab Experiment                            |
| 4                | Types of Parsers: Top down parser  | 11/01/2019          | 09/01/2019         | Class Room Teaching, Lab Experiment  |
| 5                | Design of Predictive parser  | 17/01/2019          | 11/01/2019         | Class Room Teaching  |
| 6                | Design of Predictive parser and  | 18/01/2019          | 14/01/2019         | Class Room Teaching  |
| 7                | LL(1) Grammar  | 24/01/2019          | 18/01/2019         | Class Room Teaching  |
| 8                | Design of recursive descent parser   | 25/01/2019          | 24/01/2019         | Class Room Teaching, , Lab Experiment                                      |
| 9                | Bottom up parser- Operator precedence parser   | 31/01/2019          | 25/01/2019         | Class Room Teaching  |
| 10               | Design of SLR Parser   | 1/2/2019            | 1/2/2019           | Class Room Teaching  |
| 11               | Design of LR1 parser   | 7/2/2019            | 7/2/2019           | Class Room Teaching  |
| 12               | More Examples on parsers   | 8/2/2019            | 8/2/2019           | Class Room Teaching  |
| 13               | Semantic Analysis  | 21/2/2019           | 18/2/2019          | Class Room Teaching, Self Learning Online Resources                        |
| 14               | Syntax directed definitions.   | 22/2/2019           | 18/2/2019          | Class Room Teaching, Self  |

|    |   |            |            |  |
|----|---|------------|------------|--|
|    |   |            |            | Learning Online Resources  |
| 15 | Intermediate Code Generation:<br>Types of Intermediate codes:<br>Syntax tree, Postfix notation,<br>Three address codes: Triples<br>and Quadruples.                                      | 28/2/2019  | 22/2/2019  | Class Room Teaching, Self<br>Learning Online Resources                     |
| 16 | Intermediate Code Generation:<br>examples   | 1/3/2019   | 25/2/2019  | Class Room Teaching, Self<br>Learning Online Resources ,<br>slides         |
| 17 | Code Optimization: Need and<br>sources of optimization  | 07/3/2019  | 1/3/2019   | Class Room Teaching, Self<br>Learning Online Resources                     |
| 18 | Code optimization techniques:<br>Machine Dependent  | 08/3/2019  | 6/3/2019   | Class Room Teaching, Self<br>Learning Online Resources                     |
| 19 | Code optimization techniques:<br>Machine Dependent  | 14/3/2019  | 08/3/2019  | Class Room Teaching, Self<br>Learning Online Resources                     |
| 20 | Code optimization techniques<br>:Machine Independent  | 22/3/2019  | 14/3/2019  | Class Room Teaching, Self<br>Learning Online Resources                     |
| 21 | Code Generation: Issues in the<br>design of code generator,   | 28/3/2019  | 28/3/2019  | Class Room Teaching, Lab<br>experiment , Self Learning<br>Online Resources |
| 22 | code generation algorithm.  | 29/3/2019  | 29/3/2019  | Class Room Teaching, Lab<br>experiment , Self Learning<br>Online Resources |
| 23 | Basic block and flow graph.   | 04/04/2019 | 01/04/2019 | Class Room Teaching  |
| 24 | University Paper solution   | 05/3/2019  | 03/04/2019 | Class Room Teaching  |
|    |   |            |            |  |
|    |   |            |            |  |
| 1  | Concept of System Software,<br>Goals of system softwares, system<br>program and system<br>programming,  | 1/1/19     | 1/1/19     | Class Room Teaching, Lab<br>experiment , Self Learning<br>Online Resources |
| 2  | Introduction to various system<br>programs such as Assembler,<br>Macro processor, Loader, Linker,<br>Compiler, Interpreter, Device<br>Drivers, Operating system,<br>Editors, Debuggers. | 3/1/19     | 3/1/19     | Class Room Teaching, Lab<br>experiment , Self Learning<br>Online Resources |
| 3  | Elements of Assembly Language<br>programming  | 9/1/19     | 9/1/19     | Class Room Teaching, Lab<br>experiment , Self Learning<br>Online Resources |
| 4  | Assembly scheme   | 15/1/19    | 15/1/19    | Class Room Teaching, Lab<br>experiment , Self Learning<br>Online Resources |
| 5  | pass structure of assembler   | 17/1/19    | 17/1/19    | Class Room Teaching, Lab<br>experiment , Self Learning<br>Online Resources |
| 6  | Assembler Design: Two pass<br>assembler Design and single pass<br>Assembler Design for  | 21/1/19    | 21/1/19    | Class Room Teaching, Lab<br>experiment , Self Learning<br>Online Resources |

|    |   |         |         |  |
|----|---|---------|---------|--|
|    | Hypothetical / X86 family processor, data structures used.  |         |         |  |
| 7  | Assembler Design: Two pass assembler Design and single pass Assembler Design for Hypothetical / X86 family processor, data structures used. | 22/1/19 | 22/1/19 | Class Room Teaching, Lab experiment , Self Learning Online Resources |
| 8  | Assembler Design: Two pass assembler Design and single pass Assembler Design for Hypothetical / X86 family processor, data structures used. | 28/1/19 | 28/1/19 | Class Room Teaching, Lab experiment , Self Learning Online Resources |
| 9  | Assembler Design: Two pass assembler Design and single pass Assembler Design for Hypothetical / X86 family processor, data structures used. | 29/1/19 | 29/1/19 | Class Room Teaching, Lab experiment , Self Learning Online Resources |
| 10 | Features of Macro facility: Simple, parameterized, conditional and nested   | 1/2/19  | 1/2/19  | Class Room Teaching, Lab experiment , Self Learning Online Resources |
| 11 | Features of Macro facility: Simple, parameterized, conditional and nested   | 11/2/19 | 25/2/19 | Class Room Teaching, Lab experiment , Self Learning Online Resources |
| 12 | Design of single pass macro processor, data structures used.  | 5/3/19  | 5/3/19  | Class Room Teaching, Lab experiment , Self Learning Online Resources |
| 13 | Design of single pass macro processor, data structures used.  | 11/3/19 | 12/3/19 | Class Room Teaching, Lab experiment , Self Learning Online Resources |
| 14 | Introduction, functions of loaders,   | 12/3/19 | 18/3/19 | Class Room Teaching, Lab experiment , Self Learning Online Resources |
| 15 | Relocation and Linking concept,   | 18/3/19 | 18/3/19 | Class Room Teaching, Lab experiment , Self Learning Online Resources |
| 16 | Different loading schemes: Relocating loader  | 19/3/19 | 22/3/19 | Class Room Teaching, Lab experiment , Self Learning Online Resources |
| 17 | Direct Linking Loader,  | 25/3/19 | 22/3/19 | Class Room Teaching, Lab experiment , Self Learning Online Resources |
| 18 | Dynamic linking and loading.  | 26/3/19 | 27/3/19 | Class Room Teaching, Lab experiment , Self Learning Online Resources |
| 19 | University Paper solution   | 1/4/19  | 1/4/19  | Class Room Teaching,   |
| 20 | University Paper solution   | 2/4/19  | 2/4/19  | Class Room Teaching,   |