| | | |
|---|---|---|
| LAKIREDDY BALI REDDY COLLEGE OF ENGINEERING<br>MYLAVARAM<br>TRUTH ALWAYS TRIUMPHS<br>HARD WORK PAYS | **LESSON PLAN**<br>**Sub. Name :** Object Oriented Analysis and Design<br>**Branch: CSE, Semester & Sections: V & A** | **Date:**<br>**22/06/2015**<br>**To 31/10/2015** |

## T265 – OBJECT ORIENTED ANALYSIS AND DESIGN

| | | | |
|---|---|---|---|
| **Lecture** | **: 4 Periods/week** | **Internal Marks** | **: 25** |
| **Tutorial** | **: 1** | **External Marks** | **: 75** |
| **Credits** | **: 4** | **External Examination** | **: 3 Hrs** |

----------------------------------------------------------------------------------------------------------------

### UNIT - I

**Introduction to UML:** Importance of modeling, principles of modeling, object oriented modeling, onceptual model of the UML, Architecture, and Software Development Life Cycle.

### UNIT - II

**Basic Structural Modeling:** Classes, Relationships, common Mechanisms, and diagrams.

**Advanced Structural Modeling:** Advanced classes, advanced relationships, Interfaces, Types and Roles, Packages.

### UNIT - III

**Class & Object Diagrams:** Terms, concepts, modeling techniques for Class & Object Diagrams.

### UNIT - IV

**Basic Behavioral Modeling-I :** Interactions, Interaction diagrams Use cases, Use case Diagrams, Activity Diagrams

### UNIT - V

**Advanced Behavioral Modeling:** Events and signals, state machines, processes and Threads, time and space, state chart diagrams.

**Architectural Modeling:** Component, Deployment, Component diagrams and Deployment diagrams.

### TEXT BOOK

1. Grady Booch, James Rumbaugh, Ivar Jacobson : The Unified Modeling Language User Guide, Pearson Education.

## REFERENCES

1. Meilir Page-Jones: Fundamentals of Object Oriented Design in UML, Pearson Education.
2. Pascal Roques: Modeling Software Systems Using UML2, WILEY- Dreamtech India Pvt. Ltd.
3. Atul Kahate: Object Oriented Analysis & Design, The McGraw-Hil Companies.
4. Appling UML and Patterns: An introduction to Object – Oriented Analysis and Design and Unified Process, Craig Larman, Pearson Education.

**Pre requisite**: Basic knowledge of object oriented methods, Software Engineering Concepts.

**COURSE EDUCATIONAL OBJECTIVES**:

- The main objective is the students become familiar with all phases of OOAD.
- Master the main features of the UML.
- Master the main concepts of Object Technologies and how to apply them at work and develop the ability to analyze and solve challenging problem in various domains.
- Learn the Object design Principles and understand how to apply them towards implementation.

**COURSE OUTCOMES:**

After the completion of the course, students should be able to:

1. Select the basic elements of modeling such as Things, Relationships and Diagrams depending on the views of UML Architecture and SDLC.
2. Apply basic and Advanced Structural Modeling Concepts for designing real time applications.
3. Design Class and Object Diagrams that represent Static Aspects of a Software System.
4. Analyze Dynamic Aspects of a Software System using Use Case, Interaction and Activity Diagrams.
5. Apply techniques of State Chart Diagrams and Implementation Diagrams to model behavioral aspects and Runtime environment of Software Systems.

Detailed Lesson Plan

| S.NO | DATE | TOPIC TO BE COVERED | Actual Date | No. of HOURS | Content delivery Methods |
|---|---|---|---|---|---|
| | **UNIT-I INTRODUCTION TO UML** | | | | |
| 1 | 22/06/15 | Importance of Modelling | | 1 | DM1 |
| 2 | 23/06/15 | Principles of modelling | | 1 | DM1 |
| 3 | 25/06/15 | Object Oriented modelling | | 1 | DM1 |
| 4 | 26/06/15 | Conceptual model of the UML | | 1 | DM1 |
| 5 | 27/06/15 | Conceptual model of the UML | | 1 | DM6 |
| 6 | 29/06/15 | Conceptual model of the UML | | 1 | DM6 |
| 7 | 30/06/15 | UML Architecture | | 1 | DM1 |
| 8 | 06/07/15 | Software Development life cycle | | 1 | DM6 |
| 9 | 07/07/15 | **Tutorial-I** | | 1 | **DM2** |
| 10 | 09/07/15 | Software Development life cycle | | 1 | DM1 |
| **UNIT-II STRUCTURAL MODELLING** | | | | | |
| 11 | 10/07/15 | **Basic Structural Modelling**: Classes | | 1 | DM1 |
| 12 | 13/07/15 | Basic Structural Modelling: Classes | | 1 | DM6 |
| 13 | 14/07/15 | Relation ships | | 1 | DM1 |
| 14 | 16/07/15 | Common mechanisms | | 1 | DM1 |
| 15 | 17/07/15 | Common mechanisms | | 1 | DM6 |
| 16 | 20/07/15 | Common mechanisms | | 1 | DM6 |
| 17 | 21/07/15 | Diagrams | | 1 | DM1 |
| 18 | 23/07/15 | **Tutorial-II** | | | **DM2** |
| 19 | 24/07/15 | Diagrams | | 1 | DM6 |
| 20 | 25/07/15 | **Advanced Structural Modelling: Classes** | | 1 | DM2 |
| 21 | 27/07/15 | Advanced Relation ships | | 1 | DM1 |
| 22 | 28/07/15 | Interfaces | | 1 | DM6 |
| 23 | 30/07/15 | Types and Roles | | 1 | DM6 |
| 24 | 31/07/15 | Packages | | | DM1 |
| 25 | 01/08/15 | **Tutorial-III** | | | **DM2** |

|  |  |  |  |  |  |
|---|---|---|---|---|---|
| 26 | 03/07/15 | Review |  |  | DM6 |
|  | **UNIT-III CLASSES & OBJECT DIAGRAMS** |  |  |  |  |
| 27 | 04/07/15 | Terms & Concepts - Class diagrams |  | 1 | DM1 |
| 28 | 06/07/15 | Modelling techniques for class diagrams |  | 1 | DM1 |
| 29 | 07/07/15 | Modelling techniques for class diagrams |  | 1 | DM1 |
| 30 | 10/07/15 | **MID-I EXAMS** |  |  |  |
| 31 | 11/07/15 |  |  |  |  |
| 32 | 13/07/15 |  |  |  |  |
| 33 | 14/07/15 |  |  |  |  |
| 34 | 17/08/15 |  |  |  |  |
| 35 | 18/08/15 | Terms & Concepts - Object diagrams |  | 1 | DM1 |
| 36 | 20/08/15 | Modelling techniques for Object diagrams |  | 1 | DM6 |
| 37 | 21/08/15 | Modelling techniques for Object diagrams |  | 1 | DM6 |
| 38 | 22/08/15 | **Tutorial-IV** |  | 1 | **DM2** |
|  | **UNIT-IV BASIC BEHAVIORAL MODELLING** |  |  |  |  |
| 39 | 24/08/15 | Interactions |  | 1 | DM1 |
| 40 | 25/08/15 | Interaction diagrams- Sequence |  | 1 | DM1 |
| 41 | 27/08/15 | Interaction diagrams - Collaborations |  | 1 | DM1 |
| 42 | 28/08/15 | Interaction diagrams –C M Techniques |  | 1 | DM6 |
| 43 | 29/08/15 | Use cases |  | 1 | DM1 |
| 44 | 31/08/15 | Use case Diagrams |  | 1 | DM1 |
| 45 | 01/09/15 | Common modelling techniques |  | 1 | DM6 |
| 46 | 03/09/15 | Activity diagrams |  | 1 | DM1 |
| 47 | 04/09/15 | Common modelling techniques |  | 1 | DM6 |
| 48 | 05/09/15 | **Tutorial-V** |  | 1 | **DM2** |
| 49 | 07/09/15 | Review |  |  | DM6 |
| 50 | 08/09/15 | Review |  |  | DM6 |
| **UNIT-V** |  |  |  |  |  |
| 51 | 10/09/15 | **Advanced Behavioral Modelling** |  | 1 | DM6 |
| 52 | 11/09/15 | Events and Signals |  | 1 | DM1 |

| | | | | | |
|---|---|---|---|---|---|
| 53 | 14/09/15 | Events and Signals | | 1 | DM6 |
| 54 | 15/09/15 | State machines | | 1 | DM6 |
| 55 | 18/09/15 | Processes and Threads | | 1 | DM1 |
| 56 | 19/09/15 | Time and Space | | 1 | DM1 |
| 57 | 21/09/15 | State Chart Diagrams | | 1 | DM6 |
| 58 | 22/09/15 | State Chart Diagrams | | 1 | DM6 |
| 59 | 25/09/15 | **Tutorial-VI** | | 1 | **DM2** |
| 60 | 26/09/15 | **Architectural Modelling** | | 1 | DM1 |
| 61 | 28/09/15 | Components | | 1 | DM1 |
| 62 | 29/09/15 | Component Diagrams | | 1 | DM6 |
| 63 | 01/10/15 | Deployment | | 1 | DM1 |
| 64 | 03/10/15 | Deployment Diagrams | | 1 | DM6 |
| 65 | 05/10/15 | **Tutorial-VII** | | 1 | **DM2** |
| 66 | 06/10/15 | Review | | 1 | DM6 |
| 67 | 08/10/15 | Review | | | DM6 |
| 68 | 09/10/15 | Review | | | DM6 |
| 69 | 12/10/15 | Review of Unit - IV | | | DM6 |
| 70 | 13/10/15 | Review of Unit - III | | | DM6 |
| 71 | 15/10/15 | Review of Unit - II | | | DM6 |
| 72 | 16/10/15 | Review of Unit - II | | | DM6 |
| 73 | 17/10/15 | Review of Unit - I | | | DM6 |
| 74 | 26/10/15 | **MID – II EXAMS** | | | |
| 75 | 27/10/15 | | | | |
| 76 | 29/10/15 | | | | |
| 77 | 30/10/15 | | | | |
| 78 | 31/10/15 | | | | |

TEXT BOOK :

1. Grady Booch, James Rumbaugh, Ivar Jacobson : The Unified Modeling Language User Guide, Pearson Education.

REFERENCES :

1. Meilir Page-Jones: Fundamentals of Object Oriented Design in UML, Pearson Education.
2. Atul Kahate: Object Oriented Analysis & Design, The McGraw-Hil Companies.

**NOTE: DELIVERY METHODS**: **DM1**: Lecture interspersed with discussions/BB, **DM2:** Tutorial **DM3:** Lecture with a quiz, **DM4**: Assignment/Test, **DM5**: Demonstration (laboratory, field visit) **DM6:** Presentations/PPT

At the End of the course, students attained the **Course Outcomes: CO1, CO2, CO3, CO4, CO5**, and sample proofs are enclosed in Course file.

| | | | |
|---|---|---|---|
| **Signature** | | | |
| | **Name of the Faculty** | **Name of Course**<br>**Coordinator** | **HOD** |
| | D VEERAIAH | D VEERAIAH | Dr. N RAVI SHANKAR |

| | | |
|---|---|---|
| | LESSON PLAN | Date:<br>22/06/2015<br>To 31/10/2015 |
| | Sub Code: P860<br>Sub Name: OBJECT ORIENTED ANALYSIS & DESIGN LAB<br>Branch: CSE Year:III B.Tech Semester : V | |

## P860 – OBJECT ORIENTED ANALYSIS & DESIGN LAB.

| | | | |
|---|---|---|---|
| **Lab** | **: 3 Periods/week** | **Internal Marks** | **: 25** |
| | | **External Marks** | **: 75** |
| **Credits: 2** | | **External Examination** | **: 3 Hrs** |

---

The student should take up the following case studies which are mentioned below, and Model it in different views i.e. Use case view, logical view, component view, Deployment view, Database design, forward and Reverse Engineering, and Generation of documentation of the project.

1. Unified Library application
2. Automatic Teller Machine(ATM)
3. Student Admission Procedure
4. Online Book Shopping
5. Hospital Management System
6. Cellular Network

**Pre requisite:** Object Oriented Programming concepts, Concepts of ER model.

**Course Objectives:**

To impart in depth knowledge so that the student will

1. Develop a problem statement.

2. Identify Use Cases and develop the Use Case model.

3. Identify the business activities and develop an UML Activity diagram.

4. Identity the conceptual classes and develop a domain model with UML Class diagram.

5. Be using the identified scenarios find the interaction between objects and represent those using UML

Interaction diagrams.

6. Draw the State Chart diagram.

7. Develop architecture diagram with UML package diagram notation.

8. Draw Component and Deployment diagrams.

**Course Outcomes (CO’s)**

After undergoing this laboratory module, the student should be able to:

1. Analyze Software Requirements for the given Software Application.
2. Develop the UML Diagrams to view Software System in Static and Dynamic Aspects.
3. Describe the dynamic behaviour and structure of the design.

| Session No | Topics to be covered (Week wise) | Date | Actual Date | No. of Hours | Deliver Method |
|---|---|---|---|---|---|
| 1 | Introduction to Rational Software<br>Getting Familiarity with UML Notation | 26/06/15<br>27/06/15 | | 3 | DM5&6 |
| 2 | Automatic Teller Machine(ATM) | 03/07/15<br>04/07/15 | | 3 | DM5&6 |
| 3 | Automatic Teller Machine(ATM) | 10/07/15 | | 3 | DM5&6 |
| 4 | Automatic Teller Machine(ATM) | 17/07/15 | | 3 | DM5&6 |
| 5 | Unified Library Application | 24/07/15<br>25/07/15 | | 3 | DM5&6 |
| 6 | Unified Library Application | 31/07/15<br>01/08/15 | | 3 | DM5&6 |
| 7 | Student Admission Procedure | 07/08/15 | | 3 | DM5&6 |
| 8 | Student Admission Procedure | 21/08/15<br>22/08/15 | | 3 | DM5&6 |

|  |  |  |  |  |  |
|---|---|---|---|---|---|
| 9 | Student Admission Procedure<br>Online Book Shopping | 28/08/15<br>29/08/15 |  | 3 | DM5&6 |
| 11 | Online Book Shopping | 04/09/15<br>05/09/15 |  | 3 | DM5&6 |
| 12 | Online Book Shopping | 11/09/15 |  | 3 | DM5&6 |
| 13 | Hospital Management System | 18/09/15<br>19/09/15 |  | 3 | DM5&6 |
| 14 | Cellular Network | 25/09/15<br>26/09/15 |  | 3 | DM5&6 |
| 15 | Lab Internal Exam | 03/10/15 |  | 3 | DM5&6 |
| 16 | Lab Internal Exam | 09/10/15 |  | 3 | DM5&6 |
| 17 | Revision | 16/10/15<br>17/10/15 |  | 3 | DM5&6 |

**NOTE: DELIVERY METHODS**: **DM1**: Lecture interspersed with discussions/DM1, **DM2:** Tutorial, **DM3:** Lecture with a quiz, **DM4**: Assignment/Test, **DM5**: Demonstration (laboratory, field visit), **DM6:** Presentations/PPT

At the End of the course, students attained the **Course Outcomes:** CO1, CO2, and sample proofs are enclosed in Course file.

|  |  |  |  |
|---|---|---|---|
| **Signature** |  |  |  |
|  | **Name of the Faculty** | **Name of Course Coordinator** | **HOD** |
|  | D VEERAIAH | D VEERAIAH | Dr. N RAVI SHANKAR |

| | | |
|---|---|---|
| LAKIREDDY BALI REDDY COLLEGE OF ENGINEERING<br>MYLAVARAM<br>TRUTH ALWAYS TRIUMPHS<br>HARD WORK PAYS | **LESSON PLAN** | **Date:**<br>**22/06/2015**<br>**To 31/10/2015** |
| | **Sub Code: P860**<br>**Sub Name: OBJECT ORIENTED ANALYSIS & DESIGN LAB**<br>**Branch: CSE Year:III B.Tech Semester : V (B-Sec)** | |

## P860 – OBJECT ORIENTED ANALYSIS & DESIGN LAB.

**Lab : 3 Periods/week** **Internal Marks : 25**

**External Marks : 75**

**Credits: 2** **External Examination : 3 Hrs**

---

The student should take up the following case studies which are mentioned below, and Model it in different views i.e. Use case view, logical view, component view, Deployment view, Database design, forward and Reverse Engineering, and Generation of documentation of the project.

7. Unified Library application
8. Automatic Teller Machine(ATM)
9. Student Admission Procedure
10. Online Book Shopping
11. Hospital Management System
12. Cellular Network

**Pre requisite:** Object Oriented Programming concepts, Concepts of ER model.

**Course Objectives:**

To impart in depth knowledge so that the student will

1. Develop a problem statement.

2. Identify Use Cases and develop the Use Case model.

3. Identify the business activities and develop an UML Activity diagram.

4. Identity the conceptual classes and develop a domain model with UML Class diagram.

5. Be using the identified scenarios find the interaction between objects and represent those using UML

Interaction diagrams.

6. Draw the State Chart diagram.

7. Develop architecture diagram with UML package diagram notation.

8. Draw Component and Deployment diagrams.

**Course Outcomes (CO's)**

After undergoing this laboratory module, the student should be able to:

1. Analyze Software Requirements for the given Software Application.
2. Develop the UML Diagrams to view Software System in Static and Dynamic Aspects.
3. Describe the dynamic behaviour and structure of the design.

| Session No | **Topics to be covered (Week wise)** | Date | Actual Date | No. of Hours | Delivery Methods |
|---|---|---|---|---|---|
| 1 | Introduction to Rational Software<br>Getting Familiarity with UML Notation | 22/06/15<br>24/06/15 | | 3 | DM5&6 |
| 2 | Automatic Teller Machine(ATM) | 29/06/15<br>01/07/15 | | 3 | DM5&6 |
| 3 | Automatic Teller Machine(ATM) | 06/07/15<br>08/07/15 | | 3 | DM5&6 |
| 4 | Unified Library Application | 13/07/15<br>15/07/15 | | 3 | DM5&6 |
| 5 | Unified Library Application | 20/07/15<br>22/07/15 | | 3 | DM5&6 |
| 6 | Student Admission Procedure | 27/07/15<br>29/07/15 | | 3 | DM5&6 |
| 7 | Student Admission Procedure | 03/08/15 | | 3 | DM5&6 |

| | | | | | |
|---|---|---|---|---|---|
| | | 05/08/15 | | | |
| 8 | Student Admission Procedure | 19/08/15 | | 3 | DM5&6 |
| 9 | Online Book Shopping | 24/08/15<br>26/08/15 | | 3 | DM5&6 |
| 11 | Online Book Shopping | 31/08/15<br>02/09/15 | | 3 | DM5&6 |
| 12 | Hospital Management System | 07/09/15<br>09/09/15 | | 3 | DM5&6 |
| 13 | Hospital Management System | 14/09/15<br>16/09/15 | | 3 | DM5&6 |
| 14 | Cellular Network | 21/09/15<br>23/09/15 | | 3 | DM5&6 |
| 15 | Cellular Network | 28/09/15<br>30/0915 | | 3 | DM5&6 |
| 16 | Lab Internal Exam | 05/10/15<br>07/10/15 | | 3 | DM5&6 |
| 17 | Revision | 12/10/15<br>14/10/15 | | 3 | DM5&6 |

**NOTE: DELIVERY METHODS**: **DM1**: Lecture interspersed with discussions/DM1, **DM2:** Tutorial, **DM3:** Lecture with a quiz, **DM4**: Assignment/Test, **DM5**: Demonstration (laboratory, field visit), **DM6:** Presentations/PPT

At the End of the course, students attained the **Course Outcomes:** CO1, CO2, and sample proofs are enclosed in Course file.

| **Signature** | | | |
|---|---|---|---|
| | **Name of the Faculty** | **Name of Course Coordinator** | **HOD** |
| | D VEERAIAH | D VEERAIAH | Dr. N RAVI SHANKAR |

|  |  |  |
|---|---|---|
|  | LESSON PLAN<br>Sub. Name : Object Oriented Analysis and Design<br>Branch: CSE, Semester & Sections: V & B | Date:<br>22/06/2015<br>To 31/10/2015 |

## T265 – OBJECT ORIENTED ANALYSIS AND DESIGN

|  |  |  |  |
|---|---|---|---|
| **Lecture** | **: 4 Periods/week** | **Internal Marks** | **: 25** |
| **Tutorial** | **: 1** | **External Marks** | **: 75** |
| **Credits** | **: 4** | **External Examination** | **: 3 Hrs** |

--------------------------------------------------------------------------------------------------------------

**UNIT - I**

**Introduction to UML** : Importance of modeling, principles of modeling, object oriented modeling, conceptual model of the UML, Architecture, Software Development Life Cycle.

**UNIT - II**

**Basic Structural Modeling:** Classes, Relationships, common Mechanisms, and diagrams.

**Advanced Structural Modeling:** Advanced classes, advanced relationships, Interfaces, Types and Roles, Packages.

**UNIT - III**

**Class & Object Diagrams:** Terms, concepts, modeling techniques for Class & Object Diagrams.

**UNIT - IV**

**Basic Behavioral Modeling-I :** Interactions, Interaction diagrams Use cases, Use case Diagrams, Activity Diagrams

**UNIT - V**

**Advanced Behavioral Modeling:** Events and signals, state machines, processes and Threads, time and space, state chart diagrams.

**Architectural Modeling:** Component, Deployment, Component diagrams and Deployment diagrams.

**TEXT BOOK**

1. Grady Booch, James Rumbaugh, Ivar Jacobson : The Unified Modeling Language User Guide, Pearson Education.

**REFERENCES**

1. Meilir Page-Jones: Fundamentals of Object Oriented Design in UML, Pearson Education.
2. Pascal Roques: Modeling Software Systems Using UML2, WILEY- Dreamtech India Pvt. Ltd.
3. Atul Kahate: Object Oriented Analysis & Design, The McGraw-Hil Companies.
4. Appling UML and Patterns: An introduction to Object – Oriented Analysis and Design and Unified Process, Craig Larman, Pearson Education.

**Pre requisite**: Basic knowledge of object oriented methods, Software Engineering Concepts.

**COURSE EDUCATIONAL OBJECTIVES**:

- The main objective is the students become familiar with all phases of OOAD.
- Master the main features of the UML.
- Master the main concepts of Object Technologies and how to apply them at work and develop the ability to analyze and solve challenging problem in various domains.
- Learn the Object design Principles and understand how to apply them towards implementation.

**COURSE OUTCOMES:**

After the completion of the course, students should be able to:

6. Select the basic elements of modeling such as Things, Relationships and Diagrams depending on the views of UML Architecture and SDLC.
7. Apply basic and Advanced Structural Modeling Concepts for designing real time applications.
8. Design Class and Object Diagrams that represent Static Aspects of a Software System.
9. Analyze Dynamic Aspects of a Software System using Use Case, Interaction and Activity Diagrams.
10. Apply techniques of State Chart Diagrams and Implementation Diagrams to model behavioral aspects and Runtime environment of Software Systems.

Detailed Lesson Plan

| S.NO | DATE | TOPIC TO BE COVERED | Actual Date | No. of HOURS | Content delivery Methods |
|---|---|---|---|---|---|
| | **UNIT-I INTRODUCTION TO UML** | | | | |
| 1 | 22/06/15 | Importance of Modelling | | 1 | DM1 |
| 2 | 23/06/15 | Principles of modelling | | 1 | DM1 |
| 3 | 24/06/15 | Object Oriented modelling | | 1 | DM1 |
| 4 | 25/06/15 | Conceptual model of the UML | | 1 | DM1 |
| 5 | 26/06/15 | Conceptual model of the UML | | 1 | DM6 |
| 6 | 29/06/15 | Conceptual model of the UML | | 1 | DM6 |
| 7 | 30/06/15 | UML Architecture | | 1 | DM1 |
| 8 | 06/07/15 | Software Development life cycle | | 1 | DM6 |
| 9 | 07/07/15 | **Tutorial-I** | | 1 | **DM2** |
| 10 | 08/07/15 | Software Development life cycle | | 1 | DM1 |
| **UNIT-II STRUCTURAL MODELLING** | | | | | |
| 11 | 09/07/15 | **Basic Structural Modelling**: Classes | | 1 | DM1 |
| 12 | 10/07/15 | Basic Structural Modelling: Classes | | 1 | DM6 |
| 13 | 13/07/15 | Relation ships | | 1 | DM1 |
| 14 | 14/07/15 | Common mechanisms | | 1 | DM1 |
| 15 | 16/07/15 | Common mechanisms | | 1 | DM6 |
| 16 | 17/07/15 | Common mechanisms | | 1 | DM6 |
| 17 | 20/07/15 | Diagrams | | 1 | DM1 |
| 18 | 21/07/15 | **Tutorial-II** | | | **DM2** |
| 19 | 22/07/15 | Diagrams | | 1 | DM6 |
| 20 | 23/07/15 | **Advanced Structural Modelling: Classes** | | 1 | DM2 |
| 21 | 24/07/15 | Advanced Relation ships | | 1 | DM1 |
| 22 | 27/07/15 | Interfaces | | 1 | DM6 |
| 23 | 28/07/15 | Types and Roles | | 1 | DM6 |
| 24 | 29/07/15 | Packages | | | DM1 |
| 25 | 31/08/15 | **Tutorial-III** | | | **DM2** |

|  |  |  |  |  |  |
|---|---|---|---|---|---|
| 26 | 03/07/15 | Review |  |  | DM6 |
|  | **UNIT-III CLASSES & OBJECT DIAGRAMS** |  |  |  |  |
| 27 | 04/07/15 | Terms & Concepts - Class diagrams |  | 1 | DM1 |
| 28 | 05/07/15 | Modelling techniques for class diagrams |  | 1 | DM1 |
| 29 | 06/07/15 | Modelling techniques for class diagrams |  | 1 | DM1 |
| 30 | 07/07/15 | **MID-I EXAMS** |  |  |  |
| 31 | 10/07/15 |  |  |  |  |
| 32 | 11/07/15 |  |  |  |  |
| 33 | 12/07/15 |  |  |  |  |
| 34 | 13/08/15 |  |  |  |  |
| 35 | 17/08/15 | Terms & Concepts - Object diagrams |  | 1 | DM1 |
| 36 | 18/08/15 | Modelling techniques for Object diagrams |  | 1 | DM6 |
| 37 | 19/08/15 | Modelling techniques for Object diagrams |  | 1 | DM6 |
| 38 | 20/08/15 | **Tutorial-IV** |  | 1 | **DM2** |
|  | **UNIT-IV BASIC BEHAVIORAL MODELLING** |  |  |  |  |
| 39 | 24/08/15 | Interactions |  | 1 | DM1 |
| 40 | 25/08/15 | Interaction diagrams- Sequence |  | 1 | DM1 |
| 41 | 26/08/15 | Interaction diagrams - Collaborations |  | 1 | DM1 |
| 42 | 27/08/15 | Interaction diagrams –C M Techniques |  | 1 | DM6 |
| 43 | 28/08/15 | Use cases |  | 1 | DM1 |
| 44 | 31/08/15 | Use case Diagrams |  | 1 | DM1 |
| 45 | 01/09/15 | Common modelling techniques |  | 1 | DM6 |
| 46 | 02/09/15 | Activity diagrams |  | 1 | DM1 |
| 47 | 03/09/15 | Common modelling techniques |  | 1 | DM6 |
| 48 | 04/09/15 | **Tutorial-V** |  | 1 | **DM2** |
| 49 | 07/09/15 | Review |  |  | DM6 |
| 50 | 08/09/15 | Review |  |  | DM6 |
| **UNIT-V** |  |  |  |  |  |
| 51 | 09/09/15 | **Advanced Behavioral Modelling** |  | 1 | DM6 |
| 52 | 10/09/15 | Events and Signals |  | 1 | DM1 |

|  |  |  |  |  |  |
|---|---|---|---|---|---|
| 53 | 11/09/15 | Events and Signals |  | 1 | DM6 |
| 54 | 14/09/15 | State machines |  | 1 | DM6 |
| 55 | 15/09/15 | Processes and Threads |  | 1 | DM1 |
| 56 | 16/09/15 | Time and Space |  | 1 | DM1 |
| 57 | 18/09/15 | State Chart Diagrams |  | 1 | DM6 |
| 58 | 21/09/15 | State Chart Diagrams |  | 1 | DM6 |
| 59 | 22/09/15 | **Tutorial-VI** |  | 1 | **DM2** |
| 60 | 23/09/15 | **Architectural Modelling** |  | 1 | DM1 |
| 61 | 25/09/15 | Components |  | 1 | DM1 |
| 62 | 28/09/15 | Component Diagrams |  | 1 | DM6 |
| 63 | 29/09/15 | Deployment |  | 1 | DM1 |
| 64 | 01/10/15 | Deployment Diagrams |  | 1 | DM6 |
| 65 | 05/10/15 | **Tutorial-VII** |  | 1 | **DM2** |
| 66 | 06/10/15 | Review |  | 1 | DM6 |
| 67 | 07/10/15 | Review |  |  | DM6 |
| 68 | 08/10/15 | Review |  |  | DM6 |
| 69 | 09/10/15 | Review of Unit - IV |  |  | DM6 |
| 70 | 12/10/15 | Review of Unit - III |  |  | DM6 |
| 71 | 13/10/15 | Review of Unit - II |  |  | DM6 |
| 72 | 14/10/15 | Review of Unit - II |  |  | DM6 |
| 73 | 15/10/15 | Review of Unit - I |  |  | DM6 |
| 74 | 26/10/15 | **MID – II EXAMS** |  |  |  |
| 75 | 27/10/15 |  |  |  |  |
| 76 | 29/10/15 |  |  |  |  |
| 77 | 30/10/15 |  |  |  |  |
| 78 | 31/10/15 |  |  |  |  |

TEXT BOOK :

2. Grady Booch, James Rumbaugh, Ivar Jacobson : The Unified Modeling Language User Guide, Pearson Education.

REFERENCES :

1. Meilir Page-Jones: Fundamentals of Object Oriented Design in UML, Pearson Education.
2. Atul Kahate: Object Oriented Analysis & Design, The McGraw-Hil Companies.

**NOTE: DELIVERY METHODS**: **DM1**: Lecture interspersed with discussions/BB, **DM2:** Tutorial **DM3:** Lecture with a quiz, **DM4**: Assignment/Test, **DM5**: Demonstration (laboratory, field visit) **DM6:** Presentations/PPT

At the End of the course, students attained the **Course Outcomes: CO1, CO2, CO3, CO4, CO5**, and sample proofs are enclosed in Course file.

| **Signature** | | | |
|---|---|---|---|
| | **Name of the Faculty** | **Name of Course**<br>**Coordinator** | **HOD** |
| | D. VEERAIAH | D. VEERAIAH | Dr. N. RAVI SHANKAR |

## Course Educational objectives:

- Students will have an appreciation of the history and evolution of computer graphics, both hardware and software. Assessed by written homework assignment.
- Students will have an understanding of 2D graphics and algorithms which includes line drawing, polygon filling, clipping, and transformations.
- Students will understand the concepts of and techniques used in 3D computer graphics, including viewing transformations, hierarchical modeling, color, lighting and texture mapping.
- Students will be introduced to algorithms and techniques fundamental to 3D computer graphics and will understand the relationship between the 2D and 3D versions of such algorithms.

## Course Outcomes:

This course will enable you to:

- Able to understand the graphics applications and various interactive input and output devices.
- Able to understand and draw line, circle and ellipse using algorithms and functions to implement graphic primitives
- Able to know different geometrical transformations in 2D
- Able to learn regarding 2D Coordinate transformation , viewing functions and clipping algorithms
- Able to understand the 3D display methods , geometrical transformations and coordinate transformations.

**Pre requisite:** Knowledge of Coordinate system in Mathematics.

| S.N0 | Tentative Date | Topics to be covered | Actual Date | Num. of classes | Content Delivery Methods |
|---|---|---|---|---|---|
| | | **UNIT-1** | | | |
| 1 | 22-6-15 | Introduction | 22 | 1 | DM1 |
| 2 | 23-6-15 | Algorithm | 23 | 1 | DM1/DM6 |
| 3 | 24-6-15 | Design & analysis of Algorithms | 24 | 1 | DM1/DM6 |
| 4 | 25-6-15 | Space Complexity | 25 | 1 | DM1/ DM6 |
| 5 | 27-6-15 | Time complexity | 26 | 1 | DM1/ DM6 |
| 6 | 29-6-15 | Asymptotic Notations | 27 | 1 | DM1/ DM6 |
| 7 | 30-6-15 | Tutorial-1 | 1 | 1 | DM2 |
| 8 | 1-7-15 | Divide & Conquer General method | 29 | 1 | DM1/DM6 |
| 9 | 2-7-15 | Binary Search | 30 | 1 | DM1/DM6 |
| 10 | 4-7-15 | Finding Maximun and Minimum | 2 | 1 | DM1/ DM6 |
| 11 | 6-7-15 | Example | 4 | 1 | DM1 |
| 12 | 7-7-15 | Merge sort | 6 | 1 | DM2 |
| 13 | 8-7-15 | Example | 8 | 1 | DM1 |
| 14 | 9-7-15 | Tutorial-2 | 9 | 1 | DM2 |
| | | **UNIT-II** | | | |
| 15 | 11-7-15 | Greedy Method General method | 11 | 1 | DM1/ DM6 |
| 16 | 13-7-15 | Knapsack problem | 13 | 1 | DM1/ DM6 |
| 17 | 14-7-15 | Example | 13 | | |
| 18 | 15-7-15 | Tree Vertex Splitting | 15 | 1 | DM1/ DM6 |
| 19 | 16-7-15 | Example | 16 | | |
| 20 | 20-7-15 | Job –Sequencing with deadlines | 20 | 1 | DM1/ DM6 |
| 21 | 21-7-15 | Example | 20 | 1 | DM1 |
| 22 | 22-7-15 | Tutorial-3 | 21 | 1 | DM2 |
| 23 | 23-7-15 | Minimum cost spanning tree-prims algorithm | 25 | 1 | DM1/ DM6 |
| 24 | 25-7-15 | Krushkals algorithm | 27 | 1 | DM1 |
| 25 | 27-7-15 | Optimal Storage on Tapes | 27 | 1 | DM1/ DM6 |
| 26 | 28-7-15 | Example | 29 | | |
| 27 | 29-7-15 | Optimal Merge Pattern | 30 | 1 | DM1 |
| 28 | 30-7-15 | Example | 1 | | |
| 29 | 1-8-15 | Single source Shortest path | 3 | 1 | DM1 |
| 30 | 3-8-15 | Example | 4 | 1 | DM1 |
| | | Tutorial-4 | 5 | 1 | DM2 |
| | | **UNIT – III** | | | |
| | | Dynamic Programming-General method | | 1 | DM1/ DM6 |
| | | Multistage Graph | | 1 | DM1/ DM6 |
| | | All pairs Shortest path | | 1 | DM1/ DM6 |
| 31 | 4-8-15 | Example | | 1 | DM1 |
| 32 | 5-8-15 | Single source Shortest path | | 1 | DM1/ DM6 |
| 33 | 6-8-15 | Example | | 1 | DM1 |
| 34 | 8-8-15 | Optimal Binary Search Trees | | 1 | DM1 |
| 35 | 18-8-15 | Tutorial-5 | | 1 | DM2 |

|  |  |  |  |  |  |
|---|---|---|---|---|---|
| 36 | 19-8-15 | String Editing |  | 1 | DM1 |
| 37 | 20-8-15 | 0/1 Knapsack |  | 1 | DM1 |
| 38 | 22-8-15 | Reliabilty Design |  | 1 | DM1 |
| 39 | 24-8-15 | Travelling Salesman Problem |  | 1 | DM1/ DM6 |
| 40 | 25-8-15 | Example |  | 1 | DM1 |
| 41 | 26-8-15 | Flow shop Scheduling |  | 1 | DM1/ DM6 |
| 42 | 27-8-15 | Example |  | 1 | DM1 |
| 43 | 29-8-15 | Tutorial-6 |  | 1 | DM2 |
| **UNIT – IV** |  |  |  |  |  |
| 44 | 31-8-15 | Techniques for Binary trees |  | 1 | DM1/ DM6 |
| 45 | 1-9-15 | Techniques for Graphs |  | 1 | DM1/ DM6 |
| 46 | 2-9-15 | Connected components |  | 1 | DM1/ DM6 |
| 47 | 3-9-15 | Spanning Trees |  | 1 | DM1/ DM6 |
| 45 | 7-9-15 | Bi-Connected Components |  | 1 | DM1 |
| 46 | 8-9-15 | DFS |  | 1 | DM1 |
| 47 | 9-9-15 | Tutorial-7 |  | 1 | DM2 |
| 48 | 10-9-15 | Back tracking –General method |  | 1 | DM1 |
| 49 | 12-9-15 | The 8-Queens Problem |  | 1 | DM1 |
| 50 | 14-9-15 | Sum of Subsets |  | 1 | DM1/ DM6 |
| 48 | 15-9-15 | Graph Coloring |  | 1 | DM1 |
| 49 | 16-9-15 | Hamiltonian cycle |  | 1 | DM1 |
| 50 | 17-9-15 | Knapsack problem |  | 1 | DM1 |
| 51 | 19-9-15 | Example |  | 1 | DM1 |
| 52 | 21-9-15 | Tutorial-8 |  | 1 | DM2 |
| **UNIT – V** |  |  |  |  |  |
| 53 | 22-9-15 | Branch and Bound –method |  | 1 | DM1 |
| 54 | 23-9-15 | 0/1 Knapsack Problem |  | 1 | DM1/ DM6 |
| 55 | 26-9-15 | Travelling Sales person |  | 1 | DM1/ DM6 |
| 56 | 28-9-15 | Example |  | 1 | DM1 |
| 57 | 29-9-15 | Efficiency Considerations |  | 1 | DM1/ DM6 |
| 58 | 30-9-15 | Tutorial-9 |  | 1 | DM2 |
| 59 | 1-10-15 | NP hard and NP complete- Basic concepts |  | 1 | DM1 |
| 60 | 3-10-15 | Cook's Theorem |  | 1 | DM1 |
| 61 | 5-10-15 | NP-hard Graph Problems |  | 1 | DM1 |
| 62 | 6-10-15 | NP –hard Scheduling Problem |  | 1 | DM1 |
| 63 | 7-10-15 | Example |  | 1 | DM1 |
| 64 | 8-10-15 | Some Simplified NP –hard Problems |  | 1 | DM1/ DM6 |
| 65 | 10-10-15 | Examples |  | 1 | DM1 |
| 66 | 12-10-15 | Tutorial-10 |  | 1 | DM2 |
| 67 | 13-10-15 | Revision |  | 1 | DM1/ DM6 |
| 68 | 14-10-15 | Revision |  | 1 | DM1/ DM6 |
| 69 | 15-10-15 | Revision |  | 1 | DM1/ DM6 |
| 70 | 17-10-15 | Revision |  | 1 | DM1/ DM6 |
| TOTAL |  |  |  |  |  |
| Total number of classes required to complete the syllabus |  |  |  | 66 |  |
| Total number of classes available as per Schedule |  |  |  | 70 |  |

**NOTE: DELIVERY METHODS** :**DM1**: Lecture interspersed with discussions/BB, **DM2:** Tutorial, **DM3:** Lecture with a quiz, **DM4**: Assignment/Test, **DM5**: Demonstration ( laboratory, field visit ), **DM6:** Presentations/PPT

At the End of the course, students attained the **Course Outcomes:CO1,CO2,CO3,CO4,CO5** & sample proofs are enclosed in Course file.

| **Signature** | | | |
|---|---|---|---|
| | **Name of the Faculty** | **Name of Course Co-ordinator** | **HOD** |
| | T.V. NAGARAJU | T.V. NAGARAJU | Dr. |

## Course Educational objectives:

- Students will have an appreciation of the history and evolution of computer graphics, both hardware and software. Assessed by written homework assignment.
- Students will have an understanding of 2D graphics and algorithms which includes line drawing, polygon filling, clipping, and transformations.
- Students will understand the concepts of and techniques used in 3D computer graphics, including viewing transformations, hierarchical modeling, color, lighting and texture mapping.
- Students will be introduced to algorithms and techniques fundamental to 3D computer graphics and will understand the relationship between the 2D and 3D versions of such algorithms.

## Course Outcomes:

This course will enable you to:

- Able to understand the graphics applications and various interactive input and output devices.
- Able to understand and draw line, circle and ellipse using algorithms and functions to implement graphic primitives
- Able to know different geometrical transformations in 2D
- Able to learn regarding 2D Coordinate transformation , viewing functions and clipping algorithms
- Able to understand the 3D display methods , geometrical transformations and coordinate transformations.

**Pre requisite:** Knowledge of Coordinate system in Mathematics.

| S.N0 | Tentative Date | Topics to be covered | Actual Date | Num. of classes | Content Delivery Methods |
|---|---|---|---|---|---|
| | | **UNIT-1** | | | |
| 1 | 23-6-15 | Introduction | | 1 | DM1 |
| 2 | 24-6-15 | Algorithm | | 1 | DM1/DM6 |
| 3 | 25-6-15 | Design & analysis of Algorithms | | 1 | DM1/DM6 |
| 4 | 26-6-15 | Space Complexity | | 1 | DM1/ DM6 |
| 5 | 27-6-15 | Time complexity | | 1 | DM1/ DM6 |
| 6 | 30-6-15 | Asymptotic Notations | | 1 | DM1/ DM6 |
| 7 | 1-7-15 | Tutorial-1 | | 1 | DM2 |
| 8 | 2-7-15 | Divide & Conquer General method | | 1 | DM1/DM6 |
| 9 | 3-7-15 | Binary Search | | 1 | DM1/DM6 |
| 10 | 4-7-15 | Finding Maximun and Minimum | | 1 | DM1/ DM6 |
| 11 | 7-7-15 | Example | | 1 | DM1 |
| 12 | 8-7-15 | Merge sort | | 1 | DM2 |
| 13 | 9-7-15 | Example | | 1 | DM1 |
| 14 | 10-7-15 | Tutorial-2 | | 1 | DM2 |
| | | **UNIT-II** | | | |
| 15 | 11-7-15 | Greedy Method General method | | 1 | DM1/ DM6 |
| 16 | 14-7-15 | Knapsack problem | | 1 | DM1/ DM6 |
| | 15-7-15 | Example | | | |
| 17 | 16-7-15 | Tree Vertex Splitting | | 1 | DM1/ DM6 |
| | 17-7-15 | Example | | | |
| 18 | 21-7-15 | Job –Sequencing with deadlines | | 1 | DM1/ DM6 |
| 19 | 22-7-15 | Example | | 1 | DM1 |
| 20 | 23-7-15 | Tutorial-3 | | 1 | DM2 |
| 21 | 24-7-15 | Minimum cost spanning tree-prims algorithm | | 1 | DM1/ DM6 |
| 22 | 25-7-15 | Krushkals algorithm | | 1 | DM1 |
| 23 | 28-7-15 | Optimal Storage on Tapes | | 1 | DM1/ DM6 |
| | 29-7-15 | Example | | | |
| 24 | 30-7-15 | Optimal Merge Pattern | | 1 | DM1 |
| 25 | 31-7-15 | Single source Shortest path | | 1 | DM1 |
| 26 | 1-8-15 | Example | | 1 | DM1 |
| 27 | 4-8-15 | Tutorial-4 | | 1 | DM2 |
| | | **UNIT – III** | | | |
| 28 | | Dynamic Programming-General method | | 1 | DM1/ DM6 |
| 29 | | Multistage Graph | | 1 | DM1/ DM6 |
| 30 | | All pairs Shortest path | | 1 | DM1/ DM6 |
| 31 | 5-8-15 | Example | | 1 | DM1 |
| 32 | 6-8-15 | Single source Shortest path | | 1 | DM1/ DM6 |
| 33 | 7-8-15 | Example | | 1 | DM1 |
| 34 | 8-8-15 | Optimal Binary Search Trees | | 1 | DM1 |
| 35 | 18-8-15 | Tutorial-5 | | 1 | DM2 |
| 36 | 19-8-15 | String Editing | | 1 | DM1 |

|  |  |  |  |  |  |
|---|---|---|---|---|---|
| 37 | 20-8-15 | 0/1 Knapsack |  | 1 | DM1 |
| 38 | 21-8-15 | Reliabilty Design |  | 1 | DM1 |
| 39 | 22-8-15 | Travelling Salesman Problem |  | 1 | DM1/ DM6 |
| 40 | 25-8-15 | Example |  | 1 | DM1 |
| 41 | 26-8-15 | Flow shop Scheduling |  | 1 | DM1/ DM6 |
| 42 | 27-8-15 | Example |  | 1 | DM1 |
| 43 | 28-8-15 | Tutorial-6 |  | 1 | DM2 |
| **UNIT – IV** |  |  |  |  |  |
| 44 | 29-8-15 | Techniques for Binary trees |  | 1 | DM1/ DM6 |
| 45 | 1-9-15 | Techniques for Graphs |  | 1 | DM1/ DM6 |
| 46 | 2-9-15 | Connected components |  | 1 | DM1/ DM6 |
| 47 | 3-9-15 | Spanning Trees |  | 1 | DM1/ DM6 |
| 45 | 4-9-15 | Bi-Connected Components |  | 1 | DM1 |
| 46 | 8-9-15 | DFS |  | 1 | DM1 |
| 47 | 9-9-15 | Tutorial-7 |  | 1 | DM2 |
| 48 | 10-9-15 | Back tracking –General method |  | 1 | DM1 |
| 49 | 11-9-15 | The 8-Queens Problem |  | 1 | DM1 |
| 50 | 12-9-15 | Sum of Subsets |  | 1 | DM1/ DM6 |
| 48 | 15-9-15 | Graph Coloring |  | 1 | DM1 |
| 49 | 16-9-15 | Hamiltonian cycle |  | 1 | DM1 |
| 50 | 17-9-15 | Knapsack problem |  | 1 | DM1 |
| 51 | 18-9-15 | Example |  | 1 | DM1 |
| 52 | 19-9-15 | Tutorial-8 |  | 1 | DM2 |
| **UNIT – V** |  |  |  |  |  |
| 53 | 22-9-15 | Branch and Bound –method |  | 1 | DM1 |
| 54 | 23-9-15 | 0/1 Knapsack Problem |  | 1 | DM1/ DM6 |
| 55 | 25-9-15 | Travelling Sales person |  | 1 | DM1/ DM6 |
| 56 | 26-9-15 | Example |  | 1 | DM1 |
| 57 | 29-9-15 | Efficiency Considerations |  | 1 | DM1/ DM6 |
| 58 | 30-9-15 | Tutorial-9 |  | 1 | DM2 |
| 59 | 1-10-15 | NP hard and NP complete- Basic concepts |  | 1 | DM1 |
| 60 | 3-10-15 | Cook’s Theorem |  | 1 | DM1 |
| 61 | 6-10-15 | NP-hard Graph Problems |  | 1 | DM1 |
| 62 | 7-10-15 | NP –hard Scheduling Problem |  | 1 | DM1 |
| 63 | 8-10-15 | Example |  | 1 | DM1 |
| 64 | 9-10-15 | Some Simplified NP –hard Problems |  | 1 | DM1/ DM6 |
| 65 | 10-10-15 | Examples |  | 1 | DM1 |
| 66 | 13-10-15 | Tutorial-10 |  | 1 | DM2 |
| 67 | 14-10-15 | Revision |  | 1 | DM1/ DM6 |
| 68 | 15-10-15 | Revision |  | 1 | DM1/ DM6 |
| 69 | 16-10-15 | Revision |  | 1 | DM1/ DM6 |
| 70 | 17-10-15 | Revision |  | 1 | DM1/ DM6 |
| TOTAL |  |  |  |  |  |
| Total number of classes required to complete the syllabus |  |  |  | 66 |  |
| Total number of classes available as per Schedule |  |  |  | 70 |  |

**NOTE: DELIVERY METHODS :DM1**: Lecture interspersed with discussions/BB, **DM2:** Tutorial, **DM3:** Lecture with a quiz, **DM4**: Assignment/Test, **DM5**: Demonstration ( laboratory, field visit ), **DM6:** Presentations/PPT

At the End of the course, students attained the **Course Outcomes:CO1,CO2,CO3,CO4,CO5** & sample proofs are enclosed in Course file.

| | | | |
|---|---|---|---|
| **Signature** | | | |
| | **Name of the Faculty** | **Name of Course Co-ordinator** | **HOD** |
| | T.V. NAGARAJU | T.V. NAGARAJU | Dr. |

## Lakireddy Balireddy College of Engineering (Autonomous)

**Lesson Plan: ATFL (V Sem)** ( A) **Faculty:Dr.N.Ravi Shankar** **A:Y:2015-16**

| SNo | TOPICS COVERED | Planned Date | Actual Date | Remarks |
|---|---|---|---|---|
| 1. | Fundamentals : Strings, Alphabet | 22-6-2015 | 22.6.15 | |
| 2. | Language, Operations, Finite state machine | 23-6-2015 | 23.6.15 | |
| 3. | definitions, finite automaton model' acceptance of strings, languages | 24-6-2015 | 24.6.15 | |
| 4. | deterministic finite automaton and non-deterministic finite automaton, | 25-6-2015 | 25.6.15 | |
| 5. | Transition diagrams and Language recognizers. | 26-6-2015 | 26.6.15 | |
| 6. | Finite Automata : NFA with ε transitions | 30-6-2015 | 30.6.15 | |
| 7. | Finite Automata: NFA with ε transitions - Significance, acceptance of languages. | 1-7-2015 | 1.7.15 | |
| 8. | Conversions and Equivalence : Equivalence between NFA with and without Î transitions | 2-7-2015 | 2.7.15 | |
| 9. | NFA to DFA conversion | 3-7-2015 | 3.7.15 | |
| 10. | minimization of FSM, equivalence between two FSM's, | 6-7-2015 | 6.7.15 | |
| 11. | Finite Automata with output- Moore and Mealy machines. | 7-7-2015 | 7.7.15 | |
| 12. | Regular Languages : Regular sets | 8-7-2015 | 8.7.15 | |
| 13. | regular expressions, identity rules | 9-7-2015 | 9.7.15 | |
| 14. | regular expressions, identity rules | 10-7-2015 | 10.7.15 | |
| 15. | Constructing finite Automata for a given regular expressions | 13-7-2015 | 13.7.15 | |
| 16. | Constructing finite Automata for a given regular expressions | 14-7-2015 | 14.7.15 | |
| 17. | Conversion of Finite Automata to Regular expressions | 16-7-2015 | 16.7.15 | |
| 18. | Conversion of Finite Automata to Regular expressions | 17-7-2015 | 17.7.15 | |
| 19. | Pumping lemma for regular sets | 20-7-2015 | 20.7.15 | |
| 20. | Pumping lemma for regular sets | 21-7-2015 | 21.7.15 | |
| 21. | Closure properties of regular sets | 22-7-2015 | 27.7.5 | workshop |
| 22. | Grammar Formalism | 23-7-2015 | 29.7.15 | workshop |
| 23. | Grammar Formalism | 24-7-2015 | 30.7.15 | workshop |
| 24. | Grammar Formalism | 27-7-2015 | 31.7.15 | |
| 25. | Regular grammars-right linear and left linear grammars | 29-7-2015 | 3.8.15 | |
| 26. | Regular grammars-right linear and left linear grammars | 30-7-2015 | 4.8.15 | |
| 27. | equivalence between regular linear grammar and FA | 31-7-2015 | 5.8.15 | |
| 28. | equivalence between regular linear grammar and FA | 3-8-2015 | 6.8.15 | |
| 29. | inter conversion between LLG and RLG | 4-8-2015 | 7.8.15 | |
| 30. | inter conversion between LLG and RLG | 5-8-2015 | 8.8.15 | |
| 31. | Context free grammar, | 6-8-2015 | 9.8.15 | |
| 32. | Context free grammar, | 7-8-2015 | 10.8.15 | |
| 33. | derivation trees | 10-8-2017 | 12.8.15 | |

|  |  |  |  |  |
|---|---|---|---|---|
| 34. | sentential forms, | 12-8-2015 | 13.8.15 |  |
| 35. | Right most and leftmost derivation of strings | 13-8-2015 | 13.8.15 |  |
| 36. | Context Free Grammars : Ambiguity in context free grammars, | 14-8-2015 | 14.8.15 |  |
| 37. | Context Free Grammars : Ambiguity in context free grammars, | 17-8-2015 | 17.8.15 |  |
| 38. | Minimization of Context Free Grammars | 18-8-2015 | 18.8.15 |  |
| 39. | Minimization of Context Free Grammars | 19-8-2015 | 19.8.15 |  |
| 40. | Minimization of Context Free Grammars | 20-8-2015 | 20.8.15 |  |
| 41. | Chomsky normal form | 21-8-2015 | 24.8.15 |  |
| 42. | Greibach's Normalform | 24-8-2015 | 27.8.15 |  |
| 43. | Greibach's Normalform | 25-8-2015 | 28.8.15 |  |
| 44. | Pumping Lemma for Context Free Languages. | 26-8-2015 | 28.8.15 |  |
| 45. | Pumping Lemma for Context Free Languages. | 27-8-2015 | 29.8.15 |  |
| 46. | Push Down Automata | 28-8-2015 | 29.8.15 |  |
| 47. | Push Down Automata | 31-8-2015 | 31.8.15 |  |
| 48. | model, acceptance of CFL | 1-9-2015 | 1.9.15 |  |
| 49. | Acceptance by final state and acceptance by empty state and its equivalence | 2-9-2015 | 2.9.15 |  |
| 50. | Equivalence of CFL and PDA | 3-9-2015 | 3.9.15 |  |
| 51. | Inter conversion | 4-9-2015 | 4.9.15 |  |
| 52. | Chomsky hierarchy of languages | 7-9-2015 | 7.9.15 |  |
| 53. | Chomsky hierarchy of languages and context sensitive language | 8-9-2015 | 8.9.15 |  |
| 54. | LR(0) grammar | 9-9-2015 | 9.9.15 |  |
| 55. | decidability problems, | 10-9-2015 | 10.9.15 |  |
| 56. | Turing Machine definition, | 11-9-2015 | 11.9.15 |  |
| 57. | Turing Machine construction | 14-9-2015 | 14.9.15 |  |
| 58. | Turing Machine construction | 15-9-2015 | 15.9.15 |  |
| 59. | Turing Machine construction | 16-9-2015 | 16.9.15 |  |
| 60. | Turing Machine construction | 18-9-2015 | 18.9.15 |  |
| 61. | Turing Machine construction | 21-9-2015 | 21.9.15 |  |
| 62. | Turing Machine construction | 22-9-2015 | 22.9.15 |  |
| 63. | Turing Machine construction | 23-9-2015 | 23.9.15 |  |
| 64. | Turing Machine construction | 25-9-2015 | 25.9.15 |  |
| 65. | Universal Turing Machine | 28-9-2015 | 28.9.15 |  |
| 66. | NP complete and NP hard problems | 29-9-2015 | 29.9.15 |  |
| 67. | Computable functions | 30-9-2015 | 30.9.15 |  |
| 68. | recursively enumerable languages | 1-10-2015 | 1.10.15 |  |
| 69. | Recursive languages | 5-10-2015 | 5.10.15 |  |
| 70. | Church's hypothesis | 6-10-2015 | 6.10.15 |  |
| 71. | counter machine | 7-10-2015 | 7.10.15 |  |
| 72. | types of Turing machines | 8-10-2015 | 8.10.15 |  |
| 73. | Linear Bounded Automaton | 9-10-2015 | 9.10.15 |  |
| 74. | Revision | 12-10-2015 | 12.10.15 |  |
| 75. | Revision | 13-10-2015 | 13.10.15 |  |
| 76. | Revision | 14-10-2015 | 14.10.15 |  |
| 77. | Revision | 15-10-2015 | 15.10.15 |  |
| 78. | Revision | 16-10-2015 | 16.10.15 |  |

Faculty

Head of the department

## Lakireddy Balireddy College of Engineering (Autonomous)

**Lesson Plan: ATFL (V Sem)** (B) **Faculty:Dr.N.Ravi Shankar** **A:Y:2015-16**

| SNo | TOPICS COVERED | Planned Date | Actual Date | Remarks |
|---|---|---|---|---|
| 1. | Fundamentals : Strings, Alphabet | 22-6-2015 | 22.6.15 | |
| 2. | Language, Operations, Finite state machine | 23-6-2015 | 23.6.15 | |
| 3. | definitions, finite automaton model, acceptance of strings, languages | 25-6-2015 | 25.6.15 | |
| 4. | deterministic finite automaton and non deterministic finite automaton, | 26-6-2015 | 26.6.15 | |
| 5. | Transition diagrams and Language recognizers. | 27-6-2015 | 27.7.15 | |
| 6. | Finite Automata : NFA with ε transitions | 30-6-2015 | 30.6.15 | |
| 7. | Finite Automata: NFA with ε transitions - Significance, acceptance of languages. | 2-7-2015 | 2.7.15 | |
| 8. | Conversions and Equivalence : Equivalence between NFA with and without Î transitions | 3-7-2015 | 3.7.15 | |
| 9. | NFA to DFA conversion | 4-7-2015 | 4.7.15 | |
| 10. | minimization of FSM, equivalence between two FSM's, | 6-7-2015 | 6.7.15 | |
| 11. | Finite Automata with output- Moore and Mealy machines. | 7-7-2015 | 7.7.15 | |
| 12. | Regular Languages : Regular sets | 9-7-2015 | 9.7.15 | |
| 13. | regular expressions, identity rules | 10-7-2015 | 10.7.15 | |
| 14. | regular expressions, identity rules | 11-7-2015 | 11.7.15 | |
| 15. | Constructing finite Automata for a given regular expressions | 13-7-2015 | 13.7.15 | |
| 16. | Constructing finite Automata for a given regular expressions | 14-7-2015 | 14.7.15 | |
| 17. | Conversion of Finite Automata to Regular expressions | 16-7-2015 | 16.7.15 | |
| 18. | Conversion of Finite Automata to Regular expressions | 17-7-2015 | 17.7.15 | |
| 19. | Pumping lemma for regular sets | 20-7-2015 | 20.7.15 | |
| 20. | Pumping lemma for regular sets | 21-7-2015 | 21.7.15 | |
| 21. | Closure properties of regular sets | 23-7-2015 | 25.7.15 | Workshop |
| 22. | Grammar Formalism | 24-7-2015 | 27.7.15 | Workshop |
| 23. | Grammar Formalism | 25-7-2015 | 30.7.15 | |
| 24. | Grammar Formalism | 27-7-2015 | 31.7.15 | |
| 25. | Regular grammars-right linear and left linear grammars | 30-7-2015 | 1.8.15 | |
| 26. | Regular grammars-right linear and left linear grammars | 31-7-2015 | 3.8.15 | |
| 27. | equivalence between regular linear grammar and FA | 1-8-2015 | 4.8.15 | |
| 28. | equivalence between regular linear grammar and FA | 3-8-2015 | 6.8.15 | |
| 29. | inter conversion between LLG and RLG | 4-8-2015 | 7.8.15 | |
| 30. | inter conversion between LLG and RLG | 6-8-2015 | 9.8.15 | |
| 31. | Context free grammar, | 7-8-2015 | 10.8.15 | |
| 32. | Context free grammar, | 10-8-2015 | | |
| 33. | derivation trees | 13-8-2017 | | |

|  |  |  |  |  |
|---|---|---|---|---|
| 34. | sentential forms, | 14-8-2015 | 14-8.15 |  |
| 35. | Right most and leftmost derivation of strings | 17-8-2015 | 17-8.15 |  |
| 36. | Context Free Grammars : Ambiguity in context free grammars, | 18-8-2015 | 18.8.15 |  |
| 37. | Context Free Grammars : Ambiguity in context free grammars, | 20-8-2015 | 20.8.15 |  |
| 38. | Minimization of Context Free Grammars | 21-8-2015 | 24.8.15 |  |
| 39. | Minimization of Context Free Grammars | 22-8-2015 | 24.8.15 |  |
| 40. | Minimization of Context Free Grammars | 24-8-2015 | 25-8.15 |  |
| 41. | Chomsky normal form | 25-8-2015 | 27-8.15 |  |
| 42. | Greibach's Normal form | 27-8-2015 | 28.8.15 |  |
| 43. | Greibach's Normal form | 28-8-2015 | 28.8.15 |  |
| 44. | Pumping Lemma for Context Free Languages. | 29-8-2015 | 29-8.15 |  |
| 45. | Pumping Lemma for Context Free Languages. | 31-8-2015 | 31.8.15 |  |
| 46. | Push Down Automata | 1-9-2015 | 1-9.15 |  |
| 47. | Push Down Automata | 3-9-2015 | 3.9.15 |  |
| 48. | model, acceptance of CFL | 4-9-2015 | 4-9.15 |  |
| 49. | Acceptance by final state and acceptance by empty state and its equivalence | 7-9-2015 | 7-9.15 |  |
| 50. | Equivalence of CFL and PDA | 8-9-2015 | 8.9.15 |  |
| 51. | Inter conversion | 10-9-2015 | 10.9.15 |  |
| 52. | Chomsky hierarchy of languages | 11-9-2015 | 11.9.15 |  |
| 53. | Chomsky hierarchy of languages and context sensitive language | 12-9-2015 | 12.9.15 |  |
| 54. | LR(0) grammar | 14-9-2015 | 14.9.15 |  |
| 55. | decidability problems, | 15-9-2015 | 15.9.15 |  |
| 56. | Turing Machine definition, | 18-9-2015 | 18.9.15 |  |
| 57. | Turing Machine construction | 19-9-2015 | 19.9.15 |  |
| 58. | Turing Machine construction | 21-9-2015 | 21.9.15 |  |
| 59. | Turing Machine construction | 22-9-2015 | 22.9.15 |  |
| 60. | Turing Machine construction | 25-9-2015 | 25.9.15 |  |
| 61. | Turing Machine construction | 26-9-2015 | 26.9.15 |  |
| 62. | Turing Machine construction | 28-9-2015 | 28.9.15 |  |
| 63. | Turing Machine construction | 29-9-2015 | 29.9.15 |  |
| 64. | Turing Machine construction | 1-10-2015 | 1.10.15 |  |
| 65. | Universal Turing Machine | 3-10-2015 | 3.10.15 |  |
| 66. | NP complete and NP hard problems | 5-10-2015 | 5.10.15 |  |
| 67. | Computable functions | 6-10-2015 | 6.10.15 |  |
| 68. | recursively enumerable languages | 8-10-2015 | 8.10.15 |  |
| 69. | Recursive languages | 9-10-2015 | 9.10.15 |  |
| 70. | Church's hypothesis | 10-10-2015 | 10.10.15 |  |
| 71. | counter machine | 12-10-2015 | 12.10.15 |  |
| 72. | types of Turing machines | 13-10-2015 | 13.10.15 |  |
| 73. | Linear Bounded Automaton | 15-10-2015 | 15.10.15 |  |
| 74. | Revision | 16-10-2015 | 16.10.15 |  |
| 75. | Revision | 17-10-2015 | 17.10.15 |  |

Faculty

Head of the department

LAKIREDDY BALI REDDY COLLEGE OF ENGINEERING (AUTONOMOUS)
L.B.REDDY NAGAR, MYLAVARAM, KRISHNA DIST., A.P.-521 230.
Approved by AICET, New Delhi. Accredited by NBA & Certified by ISO 9001:2008
http://www.lbrce.ac.in, cselbreddy@gmail.com, Phone: 08659-222933, Fax: 08659-222931

## DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

**Faculty Name:**,BVNR SIVA KUMAR, RH KISHAN

## LAB SCHEDULE

**Date: 26-06-2015.**

**B.Tech(v-sem-cse) A-SECTION**

| UNIT | DESCRIPTION | I Batch(SAT) | | II BATCH ( FRI ) | | |
|---|---|---|---|---|---|---|
| CYCLE | Introduction to 8086 Kits & Debug | **Planned** | **Performed** | **Planned** | **Performed** | S |
| CYCLE | Programs on Data Transfer & Exchange | 26/06/15 | | 27/06/15 | | |
| CYCLE | Programs on ADD,ADC. | 03/07/15 | | 04/07/15 | | |
| CYCLE | Programs on MUL & DIV | 10/07/15 | | 11/07/15 | | |
| CYCLE | Programs on Sorting | 17/07/15 | | | | |
| CYCLE | Programs on code Conversion | 24/07/15 | | 25/07/15 | | |
| CYCLE | Programs on String | 31/07/15 | | 01/08/15 | | |
| CYCLE | Programs on Subroutines, MASM | 07/08/15 | | 21/08/15 | | |
| CYCLE | DAC Interfacing- Generation of Waveforms | 28/08/15 | | 22/08/15 | | |
| CYCLE | ADC Interfacing | 04/09/15 | | 29/08/15 | | |
| CYCLE | Stepper Motor Interfacing | 11/09/15 | | 12/09/15 | | |
| CYCLE | Key Board Interfacing | 18/09/15 | | 19/09/15 | | |
| CYCLE | Display Interfacing | 25/09/15 | | 26/09/15 | | |
| CYCLE | 8051 Program- Program & IO | | | 03/10/15 | | |
| CYCLE | INTERNAL EXAM | 09/10/15 | | 10/10/15 | | |

# LAKIREDDY BALI REDDY COLLEGE OF ENGINEERING (AUTONOMOUS)

**L.B.REDDY NAGAR, MYLAVARAM, KRISHNA DIST., A.P.-521 230.**

**Approved by AICET, New Delhi. Accredited by NBA & Certified by ISO 9001:2008**

http://www.lbrce.ac.in, cselbreddy@gmail.com, Phone: 08659-222933, Fax: 08659-222931

## DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

**Faculty Name:**,BVNR SIVA KUMAR, RH KISHAN

## LAB SCHEDULE

**Date: 26-06-2015.**

**B.Tech(v-sem-cse) B-SECTION**

| UNIT | DESCRIPTION | I Batch(MON) | | II BATCH ( WED ) | | S |
|---|---|---|---|---|---|---|
| | | PLANED | PERFORMD | PLANED | PERFORMD | |
| CYCLE | Introduction to 8086 Kits & Debug | 07/09/15 | | 02/09/15 | | |
| CYCLE | Programs on Data Transfer & Exchange | 14/09/15 | | 09/09/15 | | |
| CYCLE | Programs on ADD,ADC. | 14/09/15 | | 16/09/15 | | |
| CYCLE | Programs on MUL & DIV | 21/09/15 | | 23/09/15 | | |
| CYCLE | Programs on Sorting | 28/09/15 | | 30/09/15 | | |
| CYCLE | Programs on code Conversion | 05/10/15 | | 07/10/15 | | |
| CYCLE | Programs on String | 12/10/15 | | 14/10/15 | | |
| CYCLE | Programs on Subroutines, MASM | 19/10/15 | | 21/10/15 | | |
| CYCLE | DAC Interfacing- Generation of Waveforms | 26/10/15 | | 28/10/15 | | |
| CYCLE | ADC Interfacing | 02/11/15 | | 04/11/15 | | |
| CYCLE | Stepper Motor Interfacing | 09/11/15 | | 11/11/15 | | |
| CYCLE | Key Board Interfacing | 16/11/15 | | 18/11/15 | | |
| CYCLE | Display Interfacing | 16/11/15 | | 25/11/15 | | |
| CYCLE | 8051 Program- Program & IO | 23/11/15 | | 25/11/15 | | |
| CYCLE | INTERNAL EXAM | 23/11/15 | | 28/10/15 | | |

LAKIREDDY BALI REDDY COLLEGE OF ENGINEERING
L.B.REDDY NAGAR, MYLAVARAM-521 230.
AUTONOMOUS & Approved by AICTE, New Delhi
Accredited by NBA of AICTE & Certified by ISO 9001:2008
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
http://www.lbrce.ac.in, ece.lbce@gmail.com Phone: 08659-222933, Fax: 08659-222931

## LESSON PLAN

**ACADEMIC YEAR : 2015-16** **COURSE: B.Tech (V-Sem)- Section**

**A**

**BRANCH : Computer Science Engineering** FACULTY : BVNR SIVA KUMAR

**SUBJECT : MICROPROCESSORS AND INTERFACING**

| S.No. | DATE | TOPIC | DATE | Teaching Methodology |
|---|---|---|---|---|
| 1 | 22/06/15 | Introduction - Microprocessor & Hexadecimal system | | |
| 2 | 23/06/15 | Architecture - Registers | | |
| 3 | 25/06/15 | Memory Segmentation, Registers usage | | |
| 4 | 26/06/15 | Instruction Format, Addressing Mode | | |
| 5 | 27/06/15 | MOV, XCHG instructions, Programs | | |
| 6 | 29/06/15 | Addressing Modes | | |
| 7 | 30/06/15 | Addressing Modes (Contd.) | | |
| 8 | 02/07/15 | Data Transfer Group, Programs | | |
| 9 | 03/07/15 | Flag Register | | |
| 10 | 04/07/15 | Data Transfer Group, Programs | | |
| 11 | 06/07/15 | ADD & ADC instructions | | |
| 12 | 07/07/15 | Arithmetic Group, programs | | |
| 13 | 09/07/15 | Arithmetic Group, programs | | |
| 14 | 10/07/15 | Tutorial - 1 | | |
| 15 | 11/07/15 | Arithmetic Group, programs | | |
| 16 | 13/07/15 | Logical Group, Programs | | |
| 17 | 14/07/15 | Logical Group, Programs | | |
| 18 | 16/07/15 | String instructions, Programs | | |
| 19 | 17/07/15 | Tutorial - 2 | | |
| 20 | 18/07/15 | Branching group, Programs | | |
| 21 | 20/07/15 | Branching group, Programs | | |
| 22 | 21/07/15 | Control group, Programs | | |
| 23 | 23/07/15 | 8086pin configuration | | |
| 24 | 24/07/15 | Tutorial - 3 | | |
| 25 | 25/07/15 | Memory interfacing | | |

| | | | | |
|---|---|---|---|---|
| 26 | 27/07/15 | Odd & Even Banks | | |
| 27 | 28/07/15 | Timing diagram | | |
| 28 | 30/07/15 | 8086 Pin Configuration | | |
| 29 | 31/07/15 | I/O interfacing | | |
| 30 | 01/08/15 | DMA Data Transfer | | |
| 31 | 03/08/15 | 8237 Block Diagram & Interfacing | | |
| 32 | 04/08/15 | 8086 Maximum Mode | | |
| 33 | 06/08/15 | Tutorial - 5 | | |
| 34 | 07/08/15 | I Mid Paper Discussion | | |
| 35 | 08/08/15 | 8255 PPI – Pin Configuration | | |
| 36 | 01/08/15 | 8255 PPI - Mode 0, Programs | | |
| 37 | 03/08/15 | DAC Interfacing, Programs | | |
| 38 | 04/08/15 | Sine wave Generation | | |

## LESSON PLAN

ACADEMIC YEAR : 2015-16 COURSE: B.Tech (VI-Sem)- Section A

BRANCH : Electronics and Communication Engineering FACULTY : BVNR SIVA KUMAR

SUBJECT : MICROPROCESSORS AND INTERFACING

| S.No. | DATE 1 | TOPIC | DATE 2 | REMARKS |
|---|---|---|---|---|
| 39 | 06/08/15 | Stepper motors & Actuators | | |
| 40 | 07/08/15 | Tutorial - 6 | | |
| 41 | 08/08/15 | Modes 1&2, ADC Interfacing | | |
| 42 | 18/08/15 | Display interfacing | | |
| 43 | 20/08/15 | Key Board interfacing | | |
| 44 | 21/08/15 | 8279 Block diagram | | |
| 45 | 22/08/15 | Interrupts, IVT | | |
| 46 | 24/08/15 | Interrupt Response, DOS Interrupts | | |
| 47 | 25/08/15 | Tutorial - 7 | | |
| 48 | 27/08/15 | PIC 8259, Modes | | |
| 49 | 28/08/15 | Interfacing & Cascading | | |
| 50 | 29/08/15 | Serial data Transfer | | |
| 51 | 31/08/15 | 8251 Architecture | | |
| 52 | 01/09/15 | Tutorial - 8 | | |
| 53 | 03/09/15 | 8251 Interfacing | | |
| 54 | 04/09/15 | Data Transfer programs, | | |
| 55 | 05/09/15 | USB | | |
| 55 | 07/09/15 | Tutorial - 9 | | |
| 56 | 08/09/15 | 8051 family specifications | | |
| 57 | 10/09/15 | Architecture | | |
| 58 | 11/09/15 | Pin configuration & Interfacing | | |
| 59 | 12/09/15 | Tutorial - 10 | | |
| 60 | 14/09/15 | Timer operation | | |
| 61 | 15/09/15 | Timer modes | | |
| 62 | 17/09/15 | Serial Port | | |
| 63 | 18/09/15 | Interrupt structure | | |
| 64 | 19/09/15 | Tutorial - 11 | | . |
| 65 | 21/09/15 | 80286 Specifications | | |
| 66 | 22/09/15 | 80386 Specifications | | |
| 67 | 24/09/15 | Real & Protected mode | | |
| 68 | 25/09/15 | Features | | |
| 69 | 26/09/15 | Memory, Paging | | |
| 70 | 28/09/15 | Tutorial - 12 | | |
| 71 | 29/09/15 | Pentium features, | | |
| 72 | 01/10/15 | Branch Prediction | | |

Signature of Faculty Signature of HOD

LAKIREDDY BALI REDDY COLLEGE OF ENGINEERING
L.B.REDDY NAGAR, MYLAVARAM-521 230.
AUTONOMOUS & Approved by AICTE, New Delhi
Accredited by NBA of AICTE & Certified by ISO 9001:2008
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
http://www.lbrce.ac.in, ece.lbce@gmail.com Phone: 08659-222933, Fax: 08659-222931

## LESSON PLAN

ACADEMIC YEAR : 2015-16 COURSE: B.Tech (V-Sem)- Section B

BRANCH : Computer Science Engineering FACULTY : BVNR SIVA KUMAR

SUBJECT : MICROPROCESSORS AND INTERFACING

| S.No. | DATE | TOPIC | DATE | Teaching Methodology |
|---|---|---|---|---|
| 1 | 22/06/15 | Introduction - Microprocessor & Hexadecimal system | | |
| 2 | 23/06/15 | Architecture - Registers | | |
| 3 | 25/06/15 | Memory Segmentation, Registers usage | | |
| 4 | 26/06/15 | Instruction Format, Addressing Mode | | |
| 5 | 27/06/15 | MOV, XCHG instructions, Programs | | |
| 6 | 29/06/15 | Addressing Modes | | |
| 7 | 30/06/15 | Addressing Modes (Contd.) | | |
| 8 | 02/07/15 | Data Transfer Group, Programs | | |
| 9 | 03/07/15 | Flag Register | | |
| 10 | 04/07/15 | Data Transfer Group, Programs | | |
| 11 | 06/07/15 | ADD & ADC instructions | | |
| 12 | 07/07/15 | Arithmetic Group, programs | | |
| 13 | 09/07/15 | Arithmetic Group, programs | | |
| 14 | 10/07/15 | Tutorial - 1 | | |
| 15 | 11/07/15 | Arithmetic Group, programs | | |
| 16 | 13/07/15 | Logical Group, Programs | | |
| 17 | 14/07/15 | Logical Group, Programs | | |
| 18 | 16/07/15 | String instructions, Programs | | |
| 19 | 17/07/15 | Tutorial - 2 | | |
| 20 | 18/07/15 | Branching group, Programs | | |
| 21 | 20/07/15 | Branching group, Programs | | |
| 22 | 21/07/15 | Control group, Programs | | |
| 23 | 23/07/15 | 8086pin configuration | | |
| 24 | 24/07/15 | Tutorial - 3 | | |
| 25 | 25/07/15 | Memory interfacing | | |
| 26 | 27/07/15 | Odd & Even Banks | | |
| 27 | 28/07/15 | Timing diagram | | |

| | | | | |
|---|---|---|---|---|
| 28 | 30/07/15 | 8086 Pin Configuration | | |
| 29 | 31/07/15 | I/O interfacing | | |
| 30 | 01/08/15 | DMA Data Transfer | | |
| 31 | 03/08/15 | 8237 Block Diagram & Interfacing | | |
| 32 | 04/08/15 | 8086 Maximum Mode | | |
| 33 | 06/08/15 | Tutorial - 5 | | |
| 34 | 07/08/15 | I Mid Paper Discussion | | |
| 35 | 08/08/15 | 8255 PPI – Pin Configuration | | |
| 36 | 01/08/15 | 8255 PPI - Mode 0, Programs | | |
| 37 | 03/08/15 | DAC Interfacing, Programs | | |
| 38 | 04/08/15 | Sine wave Generation | | |

## LESSON PLAN

ACADEMIC YEAR : 2015-16 COURSE: B.Tech (VI-Sem)- Section B

BRANCH : Electronics and Communication Engineering FACULTY : BVNR SIVA KUMAR

SUBJECT : MICROPROCESSORS AND INTERFACING

| S.No. | DATE 1 | TOPIC | DATE 2 | REMARKS |
|---|---|---|---|---|
| 39 | 06/08/15 | Stepper motors & Actuators | | |
| 40 | 07/08/15 | Tutorial - 6 | | |
| 41 | 08/08/15 | Modes 1&2, ADC Interfacing | | |
| 42 | 18/08/15 | Display interfacing | | |
| 43 | 20/08/15 | Key Board interfacing | | |
| 44 | 21/08/15 | 8279 Block diagram | | |
| 45 | 22/08/15 | Interrupts, IVT | | |
| 46 | 24/08/15 | Interrupt Response, DOS Interrupts | | |
| 47 | 25/08/15 | Tutorial - 7 | | |
| 48 | 27/08/15 | PIC 8259, Modes | | |
| 49 | 28/08/15 | Interfacing & Cascading | | |
| 50 | 29/08/15 | Serial data Transfer | | |
| 51 | 31/08/15 | 8251 Architecture | | |
| 52 | 01/09/15 | Tutorial - 8 | | |
| 53 | 03/09/15 | 8251 Interfacing | | |
| 54 | 04/09/15 | Data Transfer programs, | | |
| 55 | 05/09/15 | USB | | |
| 55 | 07/09/15 | Tutorial - 9 | | |
| 56 | 08/09/15 | 8051 family specifications | | |
| 57 | 10/09/15 | Architecture | | |
| 58 | 11/09/15 | Pin configuration & Interfacing | | |
| 59 | 12/09/15 | Tutorial - 10 | | |
| 60 | 14/09/15 | Timer operation | | |
| 61 | 15/09/15 | Timer modes | | |
| 62 | 17/09/15 | Serial Port | | |
| 63 | 18/09/15 | Interrupt structure | | |
| 64 | 19/09/15 | Tutorial - 11 | | . |
| 65 | 21/09/15 | 80286 Specifications | | |
| 66 | 22/09/15 | 80386 Specifications | | |
| 67 | 24/09/15 | Real & Protected mode | | |
| 68 | 25/09/15 | Features | | |
| 69 | 26/09/15 | Memory, Paging | | |
| 70 | 28/09/15 | Tutorial - 12 | | |
| 71 | 29/09/15 | Pentium features, | | |
| 72 | 01/10/15 | Branch Prediction | | |

Signature of Faculty Signature of HOD

| | | |
|---|---|---|
| | LESSON PLAN | Date:<br>22/06/2015<br>To 31/10/2015 |
| | Sub. Name : Principles of Programming Languages<br>Branch: CSE, Semester & Sections: V & A | |

## T284 – Principles of Programming Languages

| | | | |
|---|---|---|---|
| **Lecture** | **: 4 Periods/week** | **Internal Marks** | **: 25** |
| **Tutorial** | **: 1** | **External Marks** | **: 75** |
| **Credits** | **: 4** | **External Examination** | **: 3 Hrs** |

-----------------------------------------------------------------------------------------------------------

**UNIT - I**

**Preliminary Concepts:** The reasons for studying the concepts of programming languages, programming domains, Language evaluation criteria, influences on language design, Language categories, Programming Paradigms -- Imperative, Object Oriented, Functional programming, Logic Programming. Programming language implementation – Compilation and Virtual Machines, Programming environments

**UNIT - II**

**Syntax and Semantics:** general Problem of describing Syntax and Semantics, formal methods of describing syntax - BNF, EBNF for common programming languages features, parse trees, ambiguous grammars, attribute grammars, denotational semantics and axiomatic semantics for common programming language features.

**UNIT - III**

**Data types:** Introduction, primitive, character, user defined, array, associative, record, union, pointer and reference types, design and implementation uses related to these types. Names, Variable, concept of binding, type checking, strong typing, type compatibility, named constants, variable initialization. **Abstract DataTypes:** Abstractions and encapsulation, introductions to data abstractions, design issues, language examples, C++ parameterized ADT, object oriented programming in small talk, C++, Java, C#, Ada 95

## UNIT - IV

**Expressions and Statements:** Arithmetic relational and Boolean expressions, Short circuit evaluation mixed mode assignment, Assignment Statements, Control Structures – Statement Level, Compound Statements, Selection, Iteration, Unconditional Statements, guarded commands.

## UNIT - V

**Subprograms and Blocks:** Fundamentals of sub-programs, Scope and lifetime of variable, static and dynamic scope, Design issues of subprograms and operations, local referencing environments, parameter passing methods, overloaded sub-programs, generic sub-programs, parameters that are sub-program names, design issues for functions user defined overloaded operators, co routines. **Concurrency:** Subprogram level concurrency, semaphores, monitors, massage passing, Java threads, C# threads.

## TEXT BOOK

1. Concepts of Programming Languages Robert .W. Sebesta 6/e, Pearson Education.

## REFERENCES

1. Programming languages –Ghezzi, 3/e, John Wiley
2. Programming Languages Design and Implementation – Pratt and Zelkowitz, Fourth Edition PHI/Pearson Education
3. Programming languages –Watt, Wiley Dreamtech

**Pre requisite**: Knowledge of various Programming Languages like C, C++

**Course Objectives:**

To impart the in depth knowledge of

1. Compare programming languages;
2. Principles of programming languages design; specification of syntax and semantics
3. Describe the main principles of imperative, functional, object oriented and logic oriented programming languages;
4. Recite the high points of programming language history; and
5. Read the central formalisms used in the description of programming languages.
6. Assess programming languages critically and in a scientific manner;

**Course Outcomes:**

CO1: Master using syntax-related concepts including context-free grammars, parse trees, recursive-descent parsing, printing, and interpretation.

CO2: Master analyzing semantic issues associated with function implementations, including variable binding, scoping rules, parameter passing, and exception handling.

CO3: Master implementation techniques for interpreted functional languages.

CO4: Master using object-oriented languages.

CO5: Be familiar with design issues of object-oriented and functional languages.

CO6: Be familiar with language abstraction constructs of classes, interfaces, packages, and procedures.

CO7: Be familiar with implementation of object-oriented languages.

CO8: Be familiar with using functional languages

CO9: Be exposed to using logic languages.

Detailed Lesson Plan

| S.NO | DATE | TOPIC TO BE COVERED | Actual Date | No. of HOURS | Content delivery Methods |
|---|---|---|---|---|---|
| | **UNIT-I PRELIMINARY CONCEPTS** | | | | |
| 1 | 22/06/15 | Introduction Reasons for studying concepts of programming languages. | | 3 | DM1 |
| 2 | 25/06/15 | Programming domains. | | 1 | DM1 |
| 3 | 27/06/15 | Language Evaluation Criteria | | 1 | DM1 |
| 4 | 29/06/15 | Influences on language design | | 1 | DM1 |
| 5 | 30/06/15 | Language categories | | 1 | DM6 |
| 6 | 01/07/15 | Programming Paradigms -- Imperative, Object Oriented | | 1 | DM6 |
| 7 | 02/07/15 | Functional programming, Logic Programming. | | 1 | DM1 |
| 8 | 04/07/15 | Programming language implementation – Compilation and Virtual Machines | | 1 | DM6 |
| 9 | 06/07/15 | Programming environments | | 1 | DM1 |
| 10 | 07/07/15 | **Tutorial-I** | | 1 | **DM2** |
| **UNIT-II SYNTAX AND SEMANTICS** | | | | | |
| 11 | 08/07/15 | General Problem of describing Syntax and Semantics | | 1 | DM1 |
| 12 | 09/07/15 | BNF | | 1 | DM6 |
| 13 | 13/07/15 | Parse trees | | 1 | DM1 |
| 14 | 14/07/15 | Ambiguous grammars | | 1 | DM1 |
| 15 | 15/07/15 | Operator Precedence | | 1 | DM6 |
| 16 | 16/07/15 | Associativity | | 1 | DM6 |
| 17 | 20/07/15 | EBNF | | 1 | DM1 |
| 18 | 21/07/15 | **Tutorial-II** | | 1 | **DM2** |
| 19 | 22/07/15 | Attribute grammars | | 1 | DM6 |
| 20 | 23/07/15 | Operational | | 1 | DM2 |
| 21 | 25/07/15 | Operational, Denotation semantics and axiomatic semantics for common programming language features. | | 1 | DM1 |

| | | | | | |
|---|---|---|---|---|---|
| 22 | 27/07/15 | Operational, Denotation semantics and axiomatic semantics for common programming language features. | | 1 | |
| 23 | 28/07/15 | Operational, Denotation semantics and axiomatic semantics for common programming language features. | | 1 | |
| 24 | 29/07/15 | **Tutorial-III** | | 1 | **DM2** |
| 25 | 30/07/15 | Review | | 1 | DM6 |
| | **UNIT-III DATA TYPES** | | | | |
| 26 | 01/08/15 | **Data types:** Introduction | | 1 | DM1 |
| 27 | 03/08/15 | Primitive, character | | 1 | DM1 |
| 28 | 04/08/15 | User defined | | 1 | DM6 |
| 31 | 05/08/15 | Array Types | | 1 | DM1 |
| 32 | 06/08/15 | Array Types | | 1 | DM1 |
| 33 | 10/08/15 | **MID-I EXAMS** | | | |
| 34 | 11/08/15 | | | | |
| 35 | 12/08/15 | | | | |
| 36 | 13/08/15 | | | | |
| 37 | 17/08/15 | | | | |
| | | | | | |
| 38 | 18/08/15 | Associative Arrays | | 1 | DM1 |
| 39 | | Record Types | | | |
| 40 | 19/08/15 | Union Types | | 1 | DM6 |
| 41 | 20/08/15 | Pointer and reference types | | 1 | DM6 |
| 42 | 22/08/15 | Design and implementation uses related to above types. | | 1 | DM1 |
| 43 | 24/08/15 | Design and implementation uses related to above types. | | 1 | DM1 |
| 44 | 25/08/15 | Names, Variable, concept of binding | | 1 | DM1 |
| 45 | 26/08/15 | Type checking, strong typing, type compatibility | | 1 | DM1 |
| 46 | 27/08/15 | Named constants, variable initialization. | | 1 | DM6 |

|  |  |  |  |  |  |
|---|---|---|---|---|---|
|  | 29/08/15 | **Tutorial-III** |  | 1 | **DM2** |
| 47 | 31/08/15 | Review |  | 1 | DM1 |
| 48 | 01/09/15 | **Abstract Data Types:** Abstractions and encapsulation, introductions to data abstractions |  | 1 | DM6 |
| 49 | 02/09/15 | Design issues |  | 1 | DM1 |
| 50 | 03/09/15 | language examples |  | 1 | DM6 |
| 51 | 05/09/15 | C++ parameterized ADT |  | 1 | DM1 |
| 52 | 07/09/15 | Object oriented programming in small talk, C++, Java, C#, Ada 95 |  |  | DM6 |
| 53 | 08/09/15 | Object oriented programming in small talk, C++, Java, C#, Ada 95 |  |  | DM6 |
| 54 | 09/09/15 | Object oriented programming in small talk, C++, Java, C#, Ada 95 |  | 1 | DM6 |
|  | 10/09/15 | **Tutorial-IV** |  | 1 | **DM2** |
| 55 | 14/09/15 | **Tutorial-V** |  | 1 | **DM2** |
| 56 | 15/09/15 | Review |  | 1 | DM6 |
|  | UNIT IV **EXPRESSIONS AND STATEMENTS** |  |  |  |  |
| 57 | 16/09/15 | Arithmetic expressions |  | 1 | DM1 |
| 58 | 17/09/15 | Arithmetic expressions |  | 1 | DM1 |
| 59 | 21/09/15 | Overloaded Operators, Type Conversion |  | 1 | DM6 |
| 60 | 22/09/15 | Relational and Boolean expressions, Short circuit evaluation |  | 1 | DM6 |
| 61 | 23/09/15 | Assignment Statements, Mixed mode assignment |  | 1 | DM1 |
| 62 | 24/09/15 | Statement Level Control Statements: Selection Statements |  | 1 | DM1 |
|  | 26/09/15 | Statement Level Control Statements: Iteration Statements, Unconditional Statements, guarded commands. |  | 1 | DM1 |
| 63 | 28/09/15 | **Tutorial-VI** |  | 1 | **DM2** |
| 64 | 29/09/15 | Review |  | 1 | DM1 |
| **UNIT-V Subprograms and Blocks** |  |  |  |  |  |

| | | | | | |
|---|---|---|---|---|---|
| 65 | 30/09/15 | Fundamentals of sub-programsScope and lifetime of variable, static and dynamic Scope | | 1 | DM6 |
| 66 | 01/10/15 | Design issues of subprograms and operations, local referencing environments | | 1 | DM6 |
| 67 | 03/10/15 | Parameter passing methods, Parameters that are sub-program names. | | 1 | DM6 |
| 68 | 05/10/15 | Overloaded sub-programs, generic sub-programs, Design issues for functions, User defined overloaded operators, co routines | | 1 | DM6 |
| 69 | 06/10/15 | **Concurrency:** Subprogram level concurrency | | 1 | DM6 |
| 70 | 07/10/15 | Semaphores, Monitors | | 1 | DM6 |
| 71 | 08/10/15 | Massage passing | | 1 | DM6 |
| 72 | 10/10/15 | Java threads and C# threads | | 1 | DM6 |
| 73 | 12/10/15 | Review of Unit – V | | 1 | DM6 |
| 74 | 13/10/15 | Review of Unit – IV | | 1 | DM6 |
| 75 | 14/10/15 | Review of Unit – III | | 1 | |
| 76 | 15/10/15 | Review of Unit – II | | 1 | |
| 77 | 17/10/15 | Review of Unit – I | | 1 | |
| 78 | 26/10/15 | **MID – II EXAMS** | | | |
| 79 | 27/10/15 | | | | |
| 80 | 28/10/15 | | | | |
| 81 | 29/10/15 | | | | |
| 82 | 31/10/15 | | | | |

TEXT BOOK :

1. Concepts of Programming Languages Robert .W. Sebesta 6/e, Pearson Education.

REFERENCES :

1. Programming languages –Ghezzi, 3/e, John Wiley
2. Programming Languages Design and Implementation – Pratt and Zelkowitz, Fourth Edition PHI/Pearson Education
3. Programming languages –Watt, Wiley Dreamtech

**NOTE: DELIVERY METHODS**:

**DM1**: Lecture interspersed with discussions/BB,

**DM2:** Tutorial,
**DM3:** Lecture with a quiz,

**DM4**: Assignment/Test,

**DM5**: Demonstration (laboratory, field visit),
**DM6:** Presentations/PPT

At the End of the course, students attained the **Course Outcomes: CO1, CO2, CO3, CO4, CO5**&CO6, and sample proofs ar enclosed in Course file.

**Signature**

| | | |
|---|---|---|
| **Name of the Faculty** | **Name of Course coordinator** | **HOD** |
| T UDAYA KUMAR | | |

| | | |
|---|---|---|
| | LESSON PLAN | Date:<br>22/06/2015<br>To 31/10/2015 |
| | Sub. Name : Principles of Programming Languages<br>Branch: CSE, Semester & Sections: V & B | |

## T284 – Principles of Programming Languages

| | | | |
|---|---|---|---|
| **Lecture** | **: 4 Periods/week** | **Internal Marks** | **: 25** |
| **Tutorial** | **: 1** | **External Marks** | **: 75** |
| **Credits** | **: 4** | **External Examination** | **: 3 Hrs** |

------------------------------------------------------------------------------------------------------------

**UNIT - I**

**Preliminary Concepts:** The reasons for studying the concepts of programming languages, programming domains, Language evaluation criteria, influences on language design, Language categories, Programming Paradigms -- Imperative, Object Oriented, Functional programming, Logic Programming. Programming language implementation – Compilation and Virtual Machines, Programming environments

**UNIT - II**

**Syntax and Semantics:** general Problem of describing Syntax and Semantics, formal methods of describing syntax - BNF, EBNF for common programming languages features, parse trees, ambiguous grammars, attribute grammars, denotational semantics and axiomatic semantics for common programming language features.

**UNIT - III**

**Data types:** Introduction, primitive, character, user defined, array, associative, record, union, pointer and reference types, design and implementation uses related to these types. Names, Variable, concept of binding, type checking, strong typing, type compatibility, named constants, variable initialization. **Abstract DataTypes:** Abstractions and encapsulation, introductions to data abstractions, design issues, language examples, C++ parameterized ADT, object oriented programming in small talk, C++, Java, C#, Ada 95

## UNIT - IV

**Expressions and Statements:** Arithmetic relational and Boolean expressions, Short circuit evaluation mixed mode assignment, Assignment Statements, Control Structures – Statement Level, Compound Statements, Selection, Iteration, Unconditional Statements, guarded commands.

## UNIT - V

**Subprograms and Blocks:** Fundamentals of sub-programs, Scope and lifetime of variable, static and dynamic scope, Design issues of subprograms and operations, local referencing environments, parameter passing methods, overloaded sub-programs, generic sub-programs, parameters that are sub-program names, design issues for functions user defined overloaded operators, co routines.
**Concurrency:** Subprogram level concurrency, semaphores, monitors, massage passing, Java threads, C# threads.

## TEXT BOOK

1. Concepts of Programming Languages Robert .W. Sebesta 6/e, Pearson Education.

## REFERENCES

4. Programming languages –Ghezzi, 3/e, John Wiley
5. Programming Languages Design and Implementation – Pratt and Zelkowitz, Fourth Edition PHI/Pearson Education
6. Programming languages –Watt, Wiley Dreamtech

**Pre requisite**: Knowledge of various Programming Languages like C, C++

**Course Objectives:**

To impart the in depth knowledge of

7. Compare programming languages;
8. Principles of programming languages design; specification of syntax and semantics
9. Describe the main principles of imperative, functional, object oriented and logic oriented programming languages;
10. Recite the high points of programming language history; and

11. Read the central formalisms used in the description of programming languages.
12. Assess programming languages critically and in a scientific manner;

**Course Outcomes:**

CO1: Master using syntax-related concepts including context-free grammars, parse trees, recursive-descent parsing, printing, and interpretation.

CO2: Master analyzing semantic issues associated with function implementations, including variable binding, scoping rules, parameter passing, and exception handling.

CO3: Master implementation techniques for interpreted functional languages.

CO4: Master using object-oriented languages.

CO5: Be familiar with design issues of object-oriented and functional languages.

CO6: Be familiar with language abstraction constructs of classes, interfaces, packages, and procedures.

CO7: Be familiar with implementation of object-oriented languages.

CO8: Be familiar with using functional languages

CO9: Be exposed to using logic languages.

Detailed Lesson Plan

| S.NO | DATE | TOPIC TO BE COVERED | Actual Date | No. of HOURS | Content delivery Methods |
|---|---|---|---|---|---|
| | **UNIT-I PRELIMINARY CONCEPTS** | | | | |
| 1 | 22/06/15 | Introduction Reasons for studying concepts of programming languages. | | 3 | DM1 |
| 2 | 23/06/15 | Programming domains. | | 1 | DM1 |
| 3 | 25/06/15 | Language Evaluation Criteria | | 1 | DM1 |
| 4 | 26/06/15 | Influences on language design | | 1 | DM1 |
| 5 | 27/06/15 | Language categories | | 1 | DM6 |
| 6 | 29/06/15 | Programming Paradigms -- Imperative, Object Oriented | | 1 | DM6 |
| 7 | 30/06/15 | Functional programming, Logic Programming. | | 1 | DM1 |
| 8 | 06/07/15 | Programming language implementation – Compilation and Virtual Machines | | 1 | DM6 |
| 9 | 07/07/15 | Programming environments | | 1 | DM1 |
| 10 | 09/07/15 | **Tutorial-I** | | 1 | **DM2** |
| | **UNIT-II SYNTAX AND SEMANTICS** | | | | |
| 11 | 10/07/15 | General Problem of describing Syntax and Semantics | | 1 | DM1 |
| 12 | 13/07/15 | BNF | | 1 | DM6 |
| 13 | 14/07/15 | Parse trees | | 1 | DM1 |
| 14 | 16/07/15 | Ambiguous grammars | | 1 | DM1 |
| 15 | 17/07/15 | Operator Precedence | | 1 | DM6 |
| 16 | 20/07/15 | Associativity | | 1 | DM6 |
| 17 | 21/07/15 | EBNF | | 1 | DM1 |
| 18 | 23/07/15 | **Tutorial-II** | | 1 | **DM2** |
| 19 | 24/07/15 | Attribute grammars | | 1 | DM6 |
| 20 | 25/07/15 | Operational | | 1 | DM2 |
| 21 | 27/07/15 | Operational, Denotation semantics and axiomatic semantics for common programming language features. | | 1 | DM1 |

| | | | | | |
|---|---|---|---|---|---|
| 22 | 28/07/15 | Operational, Denotation semantics and axiomatic semantics for common programming language features. | | 1 | |
| 23 | 30/07/15 | Operational, Denotation semantics and axiomatic semantics for common programming language features. | | 1 | |
| 24 | 31/07/15 | **Tutorial-III** | | 1 | **DM2** |
| 25 | 01/08/15 | Review | | 1 | DM6 |
| | **UNIT-III DATA TYPES** | | | | |
| 26 | 03/07/15 | **Data types:** Introduction | | 1 | DM1 |
| 27 | 04/07/15 | Primitive, character | | 1 | DM1 |
| 28 | 06/07/15 | User defined | | 1 | DM6 |
| | 07/07/15 | Array Types | | 1 | DM1 |
| | 10/07/15 | **MID-I EXAMS** | | | |
| 31 | 11/07/15 | | | | |
| 32 | 13/07/15 | | | | |
| 33 | 14/07/15 | | | | |
| 34 | 17/08/15 | | | | |
| 35 | 18/08/15 | Array Types | | 1 | DM1 |
| 36 | 20/08/15 | Associative Arrays | | 1 | DM1 |
| 37 | 21/08/15 | Record Types | | | |
| 38 | 22/08/15 | Union Types | | 1 | DM6 |
| 39 | 24/08/15 | Pointer and reference types | | 1 | DM6 |
| 40 | 25/08/15 | Design and implementation uses related to above types. | | 1 | DM1 |
| 41 | 27/08/15 | Design and implementation uses related to above types. | | 1 | DM1 |
| 42 | 28/08/15 | Names, Variable, concept of binding | | 1 | DM1 |
| 43 | 29/08/15 | Type checking, strong typing, type compatibility | | 1 | DM1 |
| 44 | 31/08/15 | Named constants, variable initialization. | | 1 | DM6 |
| 45 | 01/09/15 | **Tutorial-III** | | 1 | **DM2** |

| | | | | | |
|---|---|---|---|---|---|
| 46 | 03/09/15 | Review | | 1 | DM1 |
| | | **Abstract Data Types:** Abstractions and encapsulation, introductions to data abstractions | | 1 | DM6 |
| 47 | 04/09/15 | Design issues | | 1 | DM1 |
| 48 | 05/09/15 | language examples | | 1 | DM6 |
| 49 | 07/09/15 | C++ parameterized ADT | | 1 | DM1 |
| 50 | 08/09/15 | Object oriented programming in small talk, C++, Java, C#, Ada 95 | | | DM6 |
| 51 | 10/09/15 | Object oriented programming in small talk, C++, Java, C#, Ada 95 | | | DM6 |
| 52 | 11/09/15 | Object oriented programming in small talk, C++, Java, C#, Ada 95 | | 1 | DM6 |
| 53 | 14/09/15 | **Tutorial-IV** | | 1 | **DM2** |
| 54 | 15/09/15 | **Tutorial-V** | | 1 | **DM2** |
| 55 | 18/09/15 | Review | | 1 | DM6 |
| | **UNIT-V** | | | | |
| 56 | 19/09/15 | Arithmetic expressions | | 1 | DM1 |
| 57 | 21/09/15 | Arithmetic expressions | | 1 | DM1 |
| 58 | 22/09/15 | Overloaded Operators, Type Conversion | | 1 | DM6 |
| 59 | 25/09/15 | Relational and Boolean expressions, Short circuit evaluation | | 1 | DM6 |
| 60 | 26/09/15 | Assignment Statements, Mixed mode assignment | | 1 | DM1 |
| 61 | 28/09/15 | Statement Level Control Statements: Selection Statements | | 1 | DM1 |
| 62 | 29/09/15 | Statement Level Control Statements: Iteration Statements, Unconditional Statements, guarded commands. | | 1 | DM1 |
| 63 | 01/10/15 | **Tutorial-VI** | | 1 | **DM2** |
| 64 | 03/10/15 | Review | | 1 | DM1 |
| **UNIT-V Subprograms and Blocks** | | | | | |

| | | | | | |
|---|---|---|---|---|---|
| 65 | 05/10/15 | Fundamentals of sub-programsScope and lifetime of variable, static and dynamic Scope | | 1 | DM6 |
| 66 | 06/10/15 | Design issues of subprograms and operations, local referencing environments | | 1 | DM6 |
| 67 | 08/10/15 | Parameter passing methods, Parameters that are sub-program names. | | 1 | DM6 |
| 68 | 09/10/15 | Overloaded sub-programs, generic sub-programs, Design issues for functions, User defined overloaded operators, co routines | | 1 | DM6 |
| 69 | 12/10/15 | **Concurrency:** Subprogram level concurrency | | 1 | DM6 |
| 70 | 13/10/15 | Semaphores, Monitors , Massage passing | | 1 | DM6 |
| 71 | 15/10/15 | Java threads and C# threads | | 1 | DM6 |
| 72 | 16/10/15 | Review of Unit – V AND IV | | 1 | DM6 |
| 73 | 17/10/15 | Review of Unit – III, II AND I | | 1 | DM6 |
| 74 | 26/10/15 | **MID – II EXAMS** | | | |
| 75 | 27/10/15 | | | | |
| 76 | 28/10/15 | | | | |
| 77 | 29/10/15 | | | | |
| 78 | 31/10/15 | | | | |

TEXT BOOK :

1. Concepts of Programming Languages Robert .W. Sebesta 6/e, Pearson Education.

REFERENCES :

4. Programming languages –Ghezzi, 3/e, John Wiley
5. Programming Languages Design and Implementation – Pratt and Zelkowitz, Fourth Edition PHI/Pearson Education
6. Programming languages –Watt, Wiley Dreamtech

**NOTE: DELIVERY METHODS**:

**DM1**: Lecture interspersed with discussions/BB,

**DM2:** Tutorial,
**DM3:** Lecture with a quiz,

**DM4**: Assignment/Test,

**DM5**: Demonstration (laboratory, field visit),
**DM6:** Presentations/PPT

At the End of the course, students attained the **Course Outcomes: CO1, CO2, CO3, CO4, CO5**&CO6, and sample proofs are enclosed in Course file.

**Signature**

| Name of the Faculty | Name of Course coordinator | HOD |
|---|---|---|
| T UDAYA KUMAR | | |

Name of the Faculty: **K.NAGA PRASANTHI** **Semester:1** **2015-16**

Course Title: **SOFTWARE TESTING METHODOLOGIES** **Dt:22-06-15**

| **S.No.** | **Tentative date** | **Topics to be covered** | **Actual date** | **No.of classes** | **Teaching methodology** |
|---|---|---|---|---|---|
| **1** | 22.06.2015 | 1. Purpose of Testing<br>1.1. What We Do<br>1.2. Productivity and Quality in Software<br>1.3. Goals for Testing | | **1** | BB |
| **2** | 23.06.2015 & 24.06.2015 | 1.4. Phases in a Tester's Mental Life<br>1.5. Test Design<br>1.6. Testing Isn't Everything<br>1.7. The Pesticide Paradox and the Complexity Barrier | | **2** | BB |
| **3** | 25.06.2015 & 26.06.2015 | 2. SOME DICHOTOMIES<br>2.1. Testing Versus Debugging<br>2.2. Function Versus Structure<br>2.3. The Designer Versus the Tester | | **2** | BB |
| **4** | 29.06.2015 | 2.4. Modularity Versus. Efficiency<br>2.5. Small Versus Large<br>2.6. The Builder Versus the Buyer | | **1** | BB |
| **5** | 30.06.2015 | 3. A MODEL FOR TESTING<br>3.1. The Project<br>3.2. Overview<br>3.3. The Environment<br>3.4. The Program | | **1** | BB |
| **6** | 1.07.2015 | 3.5. Bugs<br>3.6. Tests | | **1** | BB |
| **7** | 2.07.2015 | 3.7. Testing and Levels<br>3.8. The Role of Models | | **1** | BB |
| **8** | 3.07.2015 | 4. A TAXONOMY OF BUGS<br>4.1. General<br>4.2. Requirements, Features, and Functionality Bugs | | **1** | BB |
| **9** | 6.07.2015 | 4.3. Structural Bugs | | **1** | BB |
| **10** | 7.07.2015 | 4.4. Data Bugs<br>4.5. Coding Bugs | | **1** | BB |
| **11** | 8.07.2015 | 4.6. Interface, Integration, and System Bugs<br>4.7. Test and Test Design Bugs | | **1** | BB |
| **12** | 9.07.2015 | Tutorial OR Test | | **1** | BB/LCD |
| **13** | 13.07.2015 | Flow Graphs and Path testing<br>1. Predicates, Path Predicates<br>1.1. General<br>1.2. Predicates. | | **1** | **BB** |
| **14** | 14.07.2015 | 1.3. Predicate Expressions<br>1.4. Predicate Coverage<br>1.5. Testing Blindness | | **1** | BB |
| **15** | 15.07.2015 | 2. Path–Testing Basics | | **1** | BB |

| | | | | | |
|---|---|---|---|---|---|
| | | 2.1. Motivation and Assumptions<br>2.2. Control Flow graphs | | | |
| **16** | 16.07.2015 | 2.3. Path Testing | | **1** | BB |
| **17** | 17.07.2015 | 2.4. Loops<br>2.5. More on Testing Multi–Entry/Multi–Exit Routines | | **1** | BB |
| **18** | 20.07.2015 | 2.6. Effectiveness of Path Testing<br>2.7. Variations | | **1** | BB |
| **19** | 21.07.2015 | 3. Path Sensitizing<br>3.1. Review; Achievable and Unachievable Paths.<br>3.2. Pragmatic Observations | | **1** | BB |
| **20** | 22.07.2015 | 3.3. Heuristic Procedures for Sensitizing Paths<br>3.4. Examples | | **1** | BB |
| **21** | 23.07.2015 | 4. Path Instrumentation<br>4.1. The Problem<br>4.2. General Strategy<br>4.3. Link Markers | | **1** | BB |
| **22** | 24.07.2015 | 4.4. Link Counters<br>4.5. Other Instrumentation Methods.<br>4.6. Implementation | | **1** | BB |
| **23** | 27.07.2015<br>&<br>28.07.15 | 5. Application Of Path Testing<br>5.1. Integration, Coverage, and Paths in Called Components<br>5.2. New Code<br>5.3. Maintenance<br>5.4. Rehosting<br><br>6.TRANSACTION-FLOW TESTING<br>Transaction Flows<br>6.1. Definitions<br>6.2. Example<br>6.3. Usage | | **2** | BB |
| **25** | 29.07.2015 | 6.4. Implementation<br>6.5. Perspective | | **1** | |
| **26** | 30.07.2015 | 6.6. Complications<br>6.7. Transaction-Flow Structure | | **1** | BB |
| **27** | 31.07.2015 | 7. Transaction-Flow Testing Techniques<br>7.1. Get the Transaction Flows<br>7.2. Inspections, Reviews, Walkthroughs | | **1** | BB |
| **28** | 3.08.2015 | 7.3. Path Selection<br>7.4. Sensitization | | **1** | BB |
| **29** | 4.08.2015 | 7.5. Instrumentation<br>7.6. Test Databases<br>7.7. Execution | | **1** | BB |
| **30** | 5.08.2015 | DATA-FLOW TESTING<br>8. DATA-FLOW TESTING BASICS | | **1** | BB |

| | | | | | |
|---|---|---|---|---|---|
| | | 8.1. Motivation and Assumptions | | | |
| **31** | 6.08.2015 | 8.2. Date Flow-graphs<br>8.3. The Data-Flow Model | | **1** | BB |
| **32** | 7.08.2015 | 9. DATA-FLOW TESTING STRATEGIES<br>9.1. General<br>9.2. Terminology | | **1** | BB |
| **33** | 8.08.2015 | 9.3. The Strategies<br>9.4. Slicing, Dicing, Data Flow, and Debugging | | **1** | BB |
| **34** | 8.08.2015 | Tutorial\Exam\Topics beyond Syllabus | | **1** | BB/LCD |
| **35** | 18.08.2015 | DOMAIN TESTING<br>1. DOMAINS AND PATHS<br>1.1. The Model<br>1.2. A Domain Is a Set<br>1.3. Domains, Paths, and Predicates | | **1** | BB |
| **36** | 19.08.2015<br>&<br>20.08.2015 | 1.4. Domain Closure<br>1.5. Domain Dimensionality<br>1.6. The Bug Assumptions<br>1.7. Restrictions | | **2** | BB |
| **37** | 21.08.2015 | 2. NICE DOMAINS AND UGLY DOMAINS<br>2.1. Where Do Domains Come From?<br>2.2. Specified Versus Implemented Domains | | **1** | BB |
| **38** | 24.08.2015 | 2.3. Nice Domains<br>2.4. Ugly Domains and How Programmers and Testers Treat Them<br>3. DOMAIN TESTING<br>3.1. Overview<br>3.2. Domain Bugs and How to Test | | **1** | BB |
| **39** | 25.08.2015 | 3.3. Procedure<br>3.4. Variations, Tools, Effectiveness | | **1** | BB |
| **40** | 26.08.2015<br>&<br>27.08.2015 | 4. DOMAINS AND INTERFACE TESTING<br>4.1. General<br>4.2. Domains and Range<br>4.3. Closure Compatibility | | **2** | BB |
| **41** | 28.08.2015<br>&<br>31.08.2015 | 4.4. Span Compatibility<br>4.5. Interface Range/Domain Compatibility Testing<br>4.6. Finding the Values | | **2** | BB |
| **42** | 1.09.2015<br>&<br>2.09.2015 | 5. DOMAINS AND TESTABILITY<br>5.1. General<br>5.2. Linearizing Transformations | | **2** | BB |
| **43** | 3.09.2015<br>&<br>4.09.2015 | 5.3. Coordinate Transformations | | **2** | BB |

| | | | | | |
|---|---|---|---|---|---|
| | | 5.4. A Canonical Program Form<br>5.5. Great Insights? | | | |
| **44** | 7.09.2015 | Tutorial\Exam\Topics beyond Syllabus | | **1** | BB/LCD |
| **45** | 8.09.2015 | 1. PATH PRODUCTS AND PATH EXPRESSIONS<br>1.1. Overview<br>1.2. Basic Concepts<br>1.3. Path Products<br>1.4. Path Sums | | **1** | BB |
| **46** | 9.09.2015 | 1.5 Distributive Laws<br>1.6. Absorption Rule<br>1.7. Loops<br>1.8. Identity Elements | | **1** | BB |
| **47** | 10.09.2015 | 2. A REDUCTION PROCEDURE<br>2.1. Overview<br>2.2. Cross-Term Step (Step 4)<br>2.3. Parallel Term (Step 6) | | **1** | BB |
| **48** | 11.09.2015 | 2.4. Loop Term (Step 7)<br>2.5. Comments, Identities, and Node-Removal Order | | **1** | BB |
| **49** | 14.09.2015<br>&<br>15.09.2015 | 3. APPLICATIONS<br>3.1. General<br>3.2. How Many Paths in a Flowgraph?<br>3.3. Approximate Minimum Number of Paths | | **2** | BB |
| **50** | 16.09.2015<br>&<br>18.09.2015 | 3.4. The Probability of Getting There<br>3.5. The Mean Processing Time of a Routine<br>3.6. Push/Pop, Get/Return<br>3.7. Limitations and Solutions | | **2** | BB |
| **51** | 21.09.2015 | 4. REGULAR EXPRESSIONS AND FLOW-ANOMALY DETECTION<br>4.1. The Problem<br>4.2. The Method<br>4.3. A Data-Flow Testing Example<br>4.4. Generalizations, Limitations, and Comments | | **1** | BB |
| **52** | 22.09.2015 | LOGIC-BASED TESTING<br>5. MOTIVATIONAL OVERVIEW<br>5.1. Programmers and Logic<br>5.2. Hardware Logic Testing | | **1** | BB |
| **53** | 23.09.2015 | 5.3. Specification Systems and Languages 5.4. Knowledge-Based Systems<br>5.5. Overview | | **1** | BB |
| **54** | 25.09.2015 | 6. DECISION TABLES<br>6.1. Definitions and Notation<br>6.2. Decision-Table Processors<br>6.3. Decision Tables as a Basis for Test Case Design | | **1** | BB |

| | | | | | |
|---|---|---|---|---|---|
| **55** | 28.09.2015 | 6.4. Expansion of Immaterial Cases<br>6.5. Test Case Design<br>6.6. Decision Tables and Structure | | **1** | BB |
| **56** | 29.09.2015 | 7. PATH EXPRESSIONS AGAIN<br>7.1. General<br>7.2. Boolean Algebra | | **1** | BB |
| **57** | 30.09.2015 | 7.3. Boolean Equations | | **1** | |
| **58** | 1.10.2015 | 8. KV CHARTS<br>8.1. The Problem<br>8.2. Simple Forms | | **1** | BB |
| **59** | 5.10.2015 | 8.3. Three Variables<br>8.4. Four Variables and More<br>8.5. Even More Testing Strategies? | | **1** | BB |
| **60** | 6.10.2015 | 9. SPECIFICATIONS<br>9.1. General<br>9.2. Finding and Translating the Logic<br>9.3. Ambiguities and Contradictions<br>9.4. Don't-Care and Impossible Terms | | **1** | BB |
| **61** | 7.10.2015 | Tutorial\Exam\Topics beyond Syllabus | | **1** | BB |
| **62** | 8.10.2015 | 1. STATE GRAPHS<br>1.1. States<br>1.2. Inputs and Transitions<br>1.3. Outputs | | **1** | BB |
| **63** | 9.10.2015 | 1.4. State Tables<br>1.5. Time Versus Sequence<br>1.6. Software Implementation | | **1** | BB |
| **64** | 12.10.2015 | 2. GOOD STATE GRAPHS AND BAD<br>2.1. General<br>2.2. State Bugs<br>2.3. Transition Bugs<br>2.4. Output Errors<br>2.5. Encoding Bugs | | **1** | BB |
| **65** | 13.10.2015 | 3. STATE TESTING<br>3.1. Impact of Bugs<br>3.2. Principles | | **1** | BB |
| **66** | 14.10.2015 | 3.3. Limitations and Extensions<br>3.4. What to Model<br>3.5. Getting the Data<br>3.6. Tools | | **1** | BB |
| **67** | 14.10.2015 | 4. TESTABILITY TIPS<br>4.1. A Balm for Programmers<br>4.2. How Big, How Small?<br>4.3. Switches, Flags, and Unachievable Paths<br>4.4. Essential and Inessential Finite-State Behavior<br>4.5. Design Guidelines | | **1** | BB |

|  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|
| **68** | 15.10.2015 |  | 5. MOTIVATIONAL OVERVIEW<br>5.1. The Problem with Pictorial Graphs<br>5.2. Tool Building<br>5.3. Doing and Understanding Testing Theory<br>5.4. The Basic Algorithms |  | **1** | BB |
| **69** | 15.10.2015 |  | 6. THE MATRIX OF A GRAPH<br>6.1. Basic Principles<br>6.2. A Simple Weight<br>6.3. Further Notation |  | **1** | BB |
| **70** | 16.10.2015 |  | 7. RELATIONS<br>7.1. General<br>7.2. Properties of Relations<br>7.3. Equivalence Relations<br>7.4. Partial Ordering Relations<br>8. THE POWERS OF A MATRIX<br>8.1. Principles<br>8.2. Matrix Powers and Products<br>8.3. The Set of All Paths<br>8.4. Loops<br>8.5. Partitioning Algorithm<br>5.6. Breaking Loops And Applications |  | **1** | BB |
| **71** | 17.10.2015 |  | 9. NODE-REDUCTION ALGORITHM<br>9.1. General<br>9.2. Some Matrix Properties<br>9.3. The Algorithm<br>9.4. Applications<br>9.5. Some Hints |  | **1** |  |

**NOTE: DELIVERY METHODS** : **DM1**: Lecture interspersed with discussions/BB, **DM2:** Tutorial, **DM3:** Lecture with a quiz, **DM4**: Assignment/Test, **DM5**: Demonstration ( laboratory, field visit ), **DM6:** Presentations/PPT

At the End of the course, students attained the **Course Outcomes:CO1,CO2,CO3,CO4,CO5**& sample proofs are enclosed in Course file.

**Course Delivery:**

| **UNIT** | **1** |  |  | **2** |  |  |  | **3** |  | **4** |  |  |  | **5** |  |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **WEEK** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** |

|  | Prepared by | Approved by |
|---|---|---|
| Signature |  |  |
| Name | K.N. Prasanthi | HOD/CSE |
| Designation | Asistant Professor/CSE | Professor |
| *Date* |  |  |

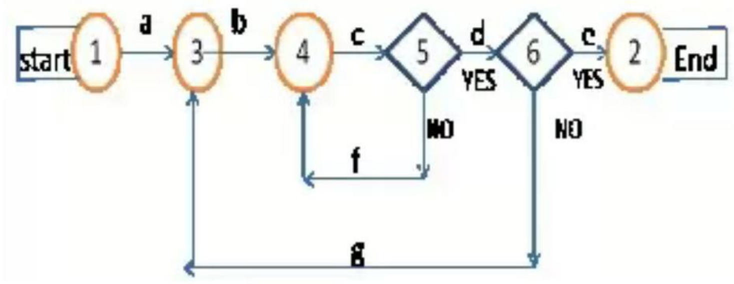
**Unit-Wise Question Bank**

**UNIT-I**

1. Why is it impossible for a tester to find all the bugs in a system? Why might it not be necessary for a program to be completely free of defects before it is delivered to its customers?
2. To what extent can testing be used to validate that the program is fit for its purpose. Discuss?
3. What is meant by integration testing? Goals of Integration Testing?
4. Explain white-box testing and behavioral testing?
5. State and explain various dichotomies in software testing?
6. Discuss about requirements, features and functionality bugs.
7. What are control and sequence bugs? How they can be caught?

**UNIT-II**

1. Consider the following flow - graph? Select optimal number of paths to achieve C1+C2 (statement coverage + branch coverage



2. Explain various loops with an example?
3. Explain concatenated loops with an example?
4. State and explain various kinds of predicate blindness with examples?

5. What are link counters? Discuss their use in path testing?
6. Discuss Traversal marker with an example?
7. What is meant by Co - incidental Correctness with example?
8. What is meant by statement testing and branch testing with an example?
9. State and explain various path selection rules.
10. What is meant by program's control flow? How is it useful for path testing?
11. Discuss various flow graph elements with their notations.
12. Distinguish Control Flow and Transaction flow.
13. What is meant by transaction flow testing? Discuss its significance.
14. Discuss in detail data - flow testing strategies.
15. What are data - flow anomalies? How data flow testing can explore them?
16. What are data-flow anomalies? How data flow testing can explore them?
17. What is meant by a program slice? Discuss about static and dynamic program slicing.
18. Explain the terms Dicing, Data-flow and Debugging.
19. What is meant by data flow model? Discuss various components of it?
20. Compare data flow and path flow testing strategies?
21. Explain data-flow testing with an example. Explain its generalizations and limitations

**UNIT-III**

1. Discuss with example the equal - span range/Doman compatibility bugs.
2. Discuss in detail about testability of Domains.
3. What is meant by Domain Dimensionality?
4. What is meant by nice - domain? Give an example for nice two - dimensional domain.
5. Discuss
    a. Linear domain boundaries
    b. Non linear domain boundaries
    c. Complete domain boundaries
    d. Incomplete domain boundaries
6. Explain various properties related to Ugly-domains.
7. State and explain various restrictions at domain testing processes.
8. What is meant by domain testing? Discuss the various applications of domain testing?
9. With a neat diagram, explain the schematic representation of domain testing.
10. Explain how one-dimensional domains are tested?
11. Discuss in detail the domains and interface testing.

**UNIT-IV**

1. Explain Regular Expressions and Flow Anomaly detection.
2. Example Huang's theorem with examples
3. Reduction procedure algorithm for the following flow graph
4. Write Short Notes on:
    a. Distributive Laws
    b. Absorption Rule
    c. Loops
    d. Identity elements
5. Discuss Path Sums and Path Product.
6. Discuss in brief applications of paths
7. Reduce the following functions using K-Maps F(A,B,C,D) = P(4,5,6,7,8,12,13)+d(1,15)
8. Whether the predicates are restricted to binary truth-values or not. Explain.

9. What are decision tables? Illustrate the applications of decision tables. How is a decision table useful in testing?
10. Explain with an example. How can we determine paths in domains in Logic based testing?
11. How the Boolean expression can be used in test case design
12. Flow graphs are abstract representations of programs. Justify?
13. Explain prime implicant, sum of product form and product of sum form.
14. How can we form specifications into sentences? Write down different phrases that can be used for words?
15. Explain about the ambiguities and contradictions in specifications.
16. Demonstrate by means of truth tables the validity of the following theorems of Boolean algebra:
    a. Associative Laws
    b. Demorgan's theorems for three variables
    c. Distributive Law
    d. Absorption Rule

**UNIT-V**

1. The behavior of a finite state machine is invariant under all encodings. Justify? (16M)**
2. Write testers comments about state graphs
3. What are the types of bugs that can cause state graphs?
4. What are the principles of state testing. Discuss advantages and disadvantages.
5. Write the design guidelines for building finite state machine into code.
6. What are the software implementation issues in state testing?
7. Explain about good state and bad state graphs.
8. Explain with an example how to convert specification into state-graph. Also discuss how contradictions can come out.
9. Write short notes on:
    a. Transition Bugs
    b. Dead States
    c. State Bugs
    d. Encoding Bugs
10. How can the graph be represented in Matrix form?
11. Write a partition algorithm.
12. Discuss node reduction algorithm.
13. How can a node reduction optimization be done.
14. What are the matrix operations in tool building.
15. Discuss the algorithm for finding set of all paths
16. How can a relation matrix be represented and what are the properties of relations? Explain cross-term reduction and node term reduction optimization.
17. Write about matrix powers and products.
18. Write about equivalence relation and partial ordering relation
19. What are the advantages and disadvantages of array representations?
20. Write about loops in matrix representation
21. What are graph matrices and their applications?
22. Discuss the linked list representation.

Course Title: **SOFTWARE TESTING METHODOLOGIES** **Dt:22-06-15**

| S.No. | Tentative date | Topics to be covered | Actual date | No.of classes | Teaching methodology |
|---|---|---|---|---|---|
| **1** | 23.06.2015 | 1. Purpose of Testing<br>1.1. What We Do<br>1.2. Productivity and Quality in Software<br>1.3. Goals for Testing | | **1** | BB |
| **2** | 24.06.2015 & 25.06.2015 | 1.4. Phases in a Tester's Mental Life<br>1.5. Test Design<br>1.6. Testing Isn't Everything<br>1.7. The Pesticide Paradox and the Complexity Barrier | | **2** | BB |
| **3** | 26.06.2015 & 27.06.2015 | 2. SOME DICHOTOMIES<br>2.1. Testing Versus Debugging<br>2.2. Function Versus Structure<br>2.3. The Designer Versus the Tester | | **2** | BB |
| **4** | 30.06.2015 | 2.4. Modularity Versus. Efficiency<br>2.5. Small Versus Large<br>2.6. The Builder Versus the Buyer | | **1** | BB |
| **5** | 1.07.2015 | 3. A MODEL FOR TESTING<br>3.1. The Project<br>3.2. Overview<br>3.3. The Environment<br>3.4. The Program | | **1** | BB |
| **6** | 2.07.2015 | 3.5. Bugs<br>3.6. Tests | | **1** | BB |
| **7** | 3.07.2015 | 3.7. Testing and Levels<br>3.8. The Role of Models | | **1** | BB |
| **8** | 4.07.2015 | 4. A TAXONOMY OF BUGS<br>4.1. General<br>4.2. Requirements, Features, and Functionality Bugs | | **1** | BB |
| **9** | 7.07.2015 | 4.3. Structural Bugs | | **1** | BB |
| **10** | 8.07.2015 | 4.4. Data Bugs<br>4.5. Coding Bugs | | **1** | BB |
| **11** | 9.07.2015 | 4.6. Interface, Integration, and System Bugs<br>4.7. Test and Test Design Bugs | | **1** | BB |
| **12** | 10.07.2015 | Tutorial OR Test | | **1** | BB/LCD |
| **13** | 11.07.2015 | Flow Graphs and Path testing<br>1. Predicates, Path Predicates<br>1.1. General<br>1.2. Predicates. | | **1** | **BB** |
| **14** | 14.07.2015 | 1.3. Predicate Expressions<br>1.4. Predicate Coverage<br>1.5. Testing Blindness | | **1** | BB |
| **15** | 15.07.2015 | 2. Path–Testing Basics<br>2.1. Motivation and Assumptions<br>2.2. Control Flow graphs | | **1** | BB |

|  |  |  |  |  |  |
|---|---|---|---|---|---|
| **16** | 16.07.2015 | 2.3. Path Testing |  | **1** | BB |
| **17** | 17.07.2015 | 2.4. Loops<br>2.5. More on Testing Multi–Entry/Multi–Exit Routines |  | **1** | BB |
| **18** | 21.07.2015 | 2.6. Effectiveness of Path Testing<br>2.7. Variations |  | **1** | BB |
| **19** | 22.07.2015 | 3. Path Sensitizing<br>3.1. Review; Achievable and Unachievable Paths.<br>3.2. Pragmatic Observations |  | **1** | BB |
| **20** | 23.07.2015 | 3.3. Heuristic Procedures for Sensitizing Paths<br>3.4. Examples |  | **1** | BB |
| **21** | 24.07.2015 | 4. Path Instrumentation<br>4.1. The Problem<br>4.2. General Strategy<br>4.3. Link Markers |  | **1** | BB |
| **22** | 25.07.2015 | 4.4. Link Counters<br>4.5. Other Instrumentation Methods.<br>4.6. Implementation |  | **1** | BB |
| **23** | 28.07.2015 | 5. Application Of Path Testing<br>5.1. Integration, Coverage, and Paths in Called Components<br>5.2. New Code<br>5.3. Maintenance<br>5.4. Rehosting<br>6.TRANSACTION-FLOW TESTING<br>Transaction Flows<br>6.1. Definitions<br>6.2. Example<br>6.3. Usage |  | **2** | BB |
| **25** | 29.07.15 | 6.4. Implementation<br>6.5. Perspective |  | **1** |  |
| **26** | 30.07.2015 | 6.6. Complications<br>6.7. Transaction-Flow Structure |  | **1** | BB |
| **27** | 31.07.2015 | 7. Transaction-Flow Testing Techniques<br>7.1. Get the Transaction Flows<br>7.2. Inspections, Reviews, Walkthroughs |  | **1** | BB |
| **28** | 1.08.2015 | 7.3. Path Selection<br>7.4. Sensitization |  | **1** | BB |
| **29** | 4.08.2015 | 7.5. Instrumentation<br>7.6. Test Databases<br>7.7. Execution |  | **1** | BB |
| **30** | 5.08.2015 | DATA-FLOW TESTING<br>8. DATA-FLOW TESTING BASICS<br>8.1. Motivation and Assumptions |  | **1** | BB |
| **31** | 6.08.2015 | 8.2. Date Flow-graphs |  | **1** | BB |

| | | | | | |
|---|---|---|---|---|---|
| | | 8.3. The Data-Flow Model | | | |
| **32** | 7.08.2015 | 9. DATA-FLOW TESTING STRATEGIES<br>9.1. General<br>9.2. Terminology | | **1** | BB |
| **33** | 8.08.2015 | 9.3. The Strategies<br>9.4. Slicing, Dicing, Data Flow, and Debugging | | **1** | BB |
| **34** | 8.08.2015 | Tutorial\Exam\Topics beyond Syllabus | | **1** | BB/LCD |
| **35** | 18.08.2015 | DOMAIN TESTING<br>1. DOMAINS AND PATHS<br>1.1. The Model<br>1.2. A Domain Is a Set<br>1.3. Domains, Paths, and Predicates | | **1** | BB |
| **36** | 19.08.2015<br>&<br>20.08.2015 | 1.4. Domain Closure<br>1.5. Domain Dimensionality<br>1.6. The Bug Assumptions<br>1.7. Restrictions | | **2** | BB |
| **37** | 21.08.2015 | 2. NICE DOMAINS AND UGLY DOMAINS<br>2.1. Where Do Domains Come From?<br>2.2. Specified Versus Implemented Domains | | **1** | BB |
| **38** | 22.08.2015 | 2.3. Nice Domains<br>2.4. Ugly Domains and How Programmers and Testers Treat Them<br>3. DOMAIN TESTING<br>3.1. Overview<br>3.2. Domain Bugs and How to Test | | **1** | BB |
| **39** | 25.08.2015 | 3.3. Procedure<br>3.4. Variations, Tools, Effectiveness | | **1** | BB |
| **40** | 26.08.2015<br>&<br>27.08.2015 | 4. DOMAINS AND INTERFACE TESTING<br>4.1. General<br>4.2. Domains and Range<br>4.3. Closure Compatibility | | **2** | BB |
| **41** | 28.08.2015<br>&<br>29.08.2015 | 4.4. Span Compatibility<br>4.5. Interface Range/Domain Compatibility Testing<br>4.6. Finding the Values | | **2** | BB |
| **42** | 1.09.2015<br>&<br>2.09.2015 | 5. DOMAINS AND TESTABILITY<br>5.1. General<br>5.2. Linearizing Transformations | | **2** | BB |
| **43** | 3.09.2015 | 5.3. Coordinate Transformations<br>5.4. A Canonical Program Form<br>5.5. Great Insights? | | **2** | BB |

| | | | | | |
|---|---|---|---|---|---|
| **44** | 4.09.2015 | Tutorial\Exam\Topics beyond Syllabus | | **1** | BB/LCD |
| **45** | 8.09.2015 | 1. PATH PRODUCTS AND PATH EXPRESSIONS<br>1.1. Overview<br>1.2. Basic Concepts<br>1.3. Path Products<br>1.4. Path Sums | | **1** | BB |
| **46** | 9.09.2015 | 1.5 Distributive Laws<br>1.6. Absorption Rule<br>1.7. Loops<br>1.8. Identity Elements | | **1** | BB |
| **47** | 10.09.2015 | 2. A REDUCTION PROCEDURE<br>2.1. Overview<br>2.2. Cross-Term Step (Step 4)<br>2.3. Parallel Term (Step 6) | | **1** | BB |
| **48** | 11.09.2015 | 2.4. Loop Term (Step 7)<br>2.5. Comments, Identities, and Node-Removal Order | | **1** | BB |
| **49** | 12.09.2015<br>&<br>15.09.2015 | 3. APPLICATIONS<br>3.1. General<br>3.2. How Many Paths in a Flowgraph?<br>3.3. Approximate Minimum Number of Paths | | **2** | BB |
| **50** | 16.09.2015 | 3.4. The Probability of Getting There<br>3.5. The Mean Processing Time of a Routine<br>3.6. Push/Pop, Get/Return<br>3.7. Limitations and Solutions | | **2** | BB |
| **51** | 18.09.2015 | 4. REGULAR EXPRESSIONS AND FLOW-ANOMALY DETECTION<br>4.1. The Problem<br>4.2. The Method<br>4.3. A Data-Flow Testing Example<br>4.4. Generalizations, Limitations, and Comments | | **1** | BB |
| **52** | 19.09.2015 | LOGIC-BASED TESTING<br>5. MOTIVATIONAL OVERVIEW<br>5.1. Programmers and Logic<br>5.2. Hardware Logic Testing | | **1** | BB |
| **53** | 22.09.2015 | 5.3. Specification Systems and Languages 5.4. Knowledge-Based Systems<br>5.5. Overview | | **1** | BB |
| **54** | 23.09.2015 | 6. DECISION TABLES<br>6.1. Definitions and Notation<br>6.2. Decision-Table Processors<br>6.3. Decision Tables as a Basis for Test Case Design | | **1** | BB |
| **55** | 25.09.2015 | 6.4. Expansion of Immaterial Cases<br>6.5. Test Case Design | | **1** | BB |

| | | | | | |
|---|---|---|---|---|---|
| | | 6.6. Decision Tables and Structure | | | |
| **56** | 26.09.2015 | 7. PATH EXPRESSIONS AGAIN<br>7.1. General<br>7.2. Boolean Algebra | | **1** | BB |
| **57** | 29.09.2015 | 7.3. Boolean Equations | | **1** | |
| **58** | 30.09.2015 | 8. KV CHARTS<br>8.1. The Problem<br>8.2. Simple Forms | | **1** | BB |
| **59** | 1.10.2015 | 8.3. Three Variables<br>8.4. Four Variables and More<br>8.5. Even More Testing Strategies? | | **1** | BB |
| **60** | 3.10.2015 | 9. SPECIFICATIONS<br>9.1. General<br>9.2. Finding and Translating the Logic<br>9.3. Ambiguities and Contradictions<br>9.4. Don't-Care and Impossible Terms | | **1** | BB |
| **61** | 6.10.2015 | Tutorial\Exam\Topics beyond Syllabus | | **1** | BB |
| **62** | 7.10.2015 | 1. STATE GRAPHS<br>1.1. States<br>1.2. Inputs and Transitions<br>1.3. Outputs | | **1** | BB |
| **63** | 8.10.2015 | 1.4. State Tables<br>1.5. Time Versus Sequence<br>1.6. Software Implementation | | **1** | BB |
| **64** | 9.10.2015 | 2. GOOD STATE GRAPHS AND BAD<br>2.1. General<br>2.2. State Bugs<br>2.3. Transition Bugs<br>2.4. Output Errors<br>2.5. Encoding Bugs | | **1** | BB |
| **65** | 10.10.2015 | 3. STATE TESTING<br>3.1. Impact of Bugs<br>3.2. Principles | | **1** | BB |
| **66** | 13.10.2015 | 3.3. Limitations and Extensions<br>3.4. What to Model<br>3.5. Getting the Data<br>3.6. Tools | | **1** | BB |
| **67** | 14.10.2015 | 4. TESTABILITY TIPS<br>4.1. A Balm for Programmers<br>4.2. How Big, How Small?<br>4.3. Switches, Flags, and Unachievable Paths<br>4.4. Essential and Inessential Finite-State Behavior<br>4.5. Design Guidelines | | **1** | BB |
| **68** | 15.10.2015 | 5. MOTIVATIONAL OVERVIEW<br>5.1. The Problem with Pictorial Graphs | | **1** | BB |

| | | | | | |
|---|---|---|---|---|---|
| | | 5.2. Tool Building<br>5.3. Doing and Understanding Testing Theory<br>5.4. The Basic Algorithms | | | |
| **69** | 15.10.2015 | 6. THE MATRIX OF A GRAPH<br>6.1. Basic Principles<br>6.2. A Simple Weight<br>6.3. Further Notation | | **1** | BB |
| **70** | 16.10.2015 | 7. RELATIONS<br>7.1. General<br>7.2. Properties of Relations<br>7.3. Equivalence Relations<br>7.4. Partial Ordering Relations<br>8. THE POWERS OF A MATRIX<br>8.1. Principles<br>8.2. Matrix Powers and Products<br>8.3. The Set of All Paths<br>8.4. Loops<br>8.5. Partitioning Algorithm<br>5.6. Breaking Loops And Applications | | **1** | BB |
| **71** | 17.10.2015 | 9. NODE-REDUCTION ALGORITHM<br>9.1. General<br>9.2. Some Matrix Properties<br>9.3. The Algorithm<br>9.4. Applications<br>9.5. Some Hints | | **1** | |

**NOTE: DELIVERY METHODS** : **DM1**: Lecture interspersed with discussions/BB, **DM2:** Tutorial, **DM3:** Lecture with a quiz, **DM4**: Assignment/Test, **DM5**: Demonstration ( laboratory, field visit ), **DM6:** Presentations/PPT

At the End of the course, students attained the **Course Outcomes:CO1,CO2,CO3,CO4,CO5**& sample proofs are enclosed in Course file.

**Course Delivery:**

| **UNIT** | **1** | | | **2** | | | | **3** | | **4** | | | | **5** | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **WEEK** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** |

| | **Prepared by** | **Approved by** |
|---|---|---|
| **Signature** | | |

| | | |
|---|---|---|
| **Name** | **K.N. Prasanthi** | **HOD/CSE** |
| **Designation** | **Asistant Professor/CSE** | **Professor** |
| *Date* | | |