Computer Graphics (3-1-3)

Evaluation:

	Theory	Practical	Total
Sessional	30	20	50
Final	50	-	50
Total	80	20	100

Course Objectives:

1. To provide the knowledge of basic techniques used in Computer Graphics Systems.
2. To provide the knowledge of 2D and 3D algorithms used in Computer Graphics Systems.

Course Contents:

Unit 1: Introduction **(2 hrs)**

1.1 Introduction
1.2 History of Computer Graphics
1.3 Application of Computer Graphics

Unit 2: Graphics Hardware **(6 hrs)**

2.1 Interactive Input Devices
2.2 Display Devices and Hard Copy Devices
2.3 Raster and Random Systems and Architectures
2.4 Video Controller
2.5 Use of Digital to Analog Converter and Frame Buffer Organization
2.6 Color Monitors

Unit 3: Two Dimensional Algorithms **(7 hrs)**

3.1 Line Drawing Algorithms
3.1.1 DDA
3.1.2 Bresenham's Algorithm
3.2 Circle Generation Algorithm
3.3 Ellipse Generation Algorithms
3.4 Area Filling-Scan Line Algorithm
3.5 Boundary Fill Techniques
3.6 Flood Fill Techniques

Unit 4: Two Dimensional Geometric Transformations and Viewing **(8 hrs)**

4.1 Basic Transformations
4.2 Other Transformations
4.3 Homogeneous Co-ordinate systems
4.4 Composite Transformations
4.5 Windowing Concepts
4.6 Viewing Pipeline
4.7 Window to View port Transformation

4.8 Line Clipping Algorithm: Cohen-Sutherland
4.9 Polygon Clipping: Sutherland-Hodgeman

Unit 5: Three Dimensional Graphics Systems **(7 hrs)**

5.1 3D Co-ordinate System and 3D Transformations
5.2 3D Representations
5.3 Polygon Surfaces
5.4 Cubic Spline and Beizer Curve
5.5 Non-Planer Surface: Bezier Surface
5.6 Fractal Geometry Method
5.7 3D Viewing Transformation
5.8 Projection Methods: Parallel and Perspective
5.9 Clipping in 3D

Unit 6: Visible Surface Detection **(5 hrs)**

6.1 Hidden Surfaces and their Removal Techniques
6.2 Back-Face Detection
6.3 Depth Buffer Method
6.4 A- buffer method
6.5 Scan Line Method
6.6 Area Subdivision Method
6.7 Depth Sorting Method

Unit 7: Illumination and Shading **(6 hrs)**

7.1 Illumination Theory
7.2 Ambient Light
7.3 Reflections: Diffuse, Specular
7.4 Surface Shading methods
  7.4.1 Constant Shading
  7.4.2 Gouraud Shading
  7.4.3 Phong Shading
  7.4.4 Fast Phong Shading
7.5 Color Models: RGB, CMYK

Unit 8: Graphical Languages **(4 hrs)**

8.1 Need for Machine Independent Graphical Languages
8.2 Graphical Languages: PHIGS, GKS
8.3 Graphics Software Standard
8.4 Overview of Graphics File Formats
8.5 Data Structure in Computer Graphics
8.6 Introduction to OpenGL

Laboratory:

Implementation of various 2D and 3D graphics algorithms covered in the course using C / C++ and OpenGL.

Text Book:

Donald Hearn and M. Pauline Baker: *Computer Graphics*. Prentice-Hall.

Reference Books:

1. James D. Foley, Andries van Dam, Steven K. Feiner, John F. Hughes, *Computer Graphics: Principles and Practice in C*, Addison-Wesley.
2. Mason Woo, Jackie Neider, Tom Davis, Dave Shreiner, *Open GL Programming Guide* Third Edition, The Official Guide to Learning OpenGL, Version 1.2, OpenGL Architecture Review Board, LPE Pearson Edition Asia.

Computer Organizations and Architecture (3-1-2)

Evaluation:

	Theory	Practical	Total
Sessional	30	20	50
Final	50	-	50
Total	80	20	100

Course Objectives:

Undergoing this course will help a student to build up a sound background in understanding the fundamentals of organization of the Computer System and the associated components. This course exposes a student to the modem trends and technology behind computer organization in a practical perspective with examples taken from real world.

Course Contents:

1 Instruction Set Architecture. **(2 hrs)**

- 1.1 Levels of Programming Language
- 1.2 Language Category, Compiling and Assembling Programs
- 1.3 Assembly Language Instructions
- 1.4 Instruction Type, Data Types, Addressing Modes, Instruction Formats
- 1.5 Instruction Set Architecture Design

2 Computer Organization **(6 hrs)**

- 2.1 Basic Computer Organization
- 2.2 System Buses
- 2.3 Instruction Cycles
- 2.4 CPU Organization
- 2.5 Memory Sub-system Organization and Interfacing
- 2.6 I/O Sub-system Organization and interfacing

3 RTL and HDL **(4 hrs)**

- 3.1 Micro-Operations and RTL
- 3.2 Using RTL to specify a Digital System
- 3.3 Specification of Digital Component,
- 3.4 Specification and Implementation of Simple System.
- 3.5 Introduction to VHDL: Syntax, Levels of Abstraction in Design

4 CPU Design **(7 hrs)**

- 4.1 Specification of a CPU
- 4.2 Design and Implementation of a Very Simple and Relatively Simple CPU
- 4.3 Instruction Execution, Fetch, Decode, Data Path
- 4.4 ALU Design
- 4.5 Designing Hardwired Control Unit
- 4.6 Design Verification

5 Control Unit Design **(4 hrs)**

5.1 Basic Micro-sequencer (Control Unit) Design and Operations
5.2 Micro-instruction Formats
5.3 Design and Implementation of a Very Simple Micro-sequencer
5.4 Control Unit: Layout, Control Sequence Generation, Mapping Logic
5.5 Generation of Micro-Operations using Horizontal and Vertical Microcode
5.6 Directly Generating the Control Signals from the Microcode
5.7 Reducing tile Number of Micro-Instructions
5.8 Micro-programmed vs. Hardwired Control Unit

6 Arithmetic Unit **(6 hrs)**

6.1 Representations of Binary Number and Arithmetic in Unsigned Notation
6.2 Addition and Subtraction in Unsigned Notation
6.3 Multiplication in Unsigned Notation, Shift Add Multiplication Algorithm, Booth's Algorithm
6.4 Division in Unsigned Notation, Shift Subtract Division Algorithm
6.5 Signed Notation
6.6 Addition and Subtraction in Signed Notation
6.7 Binary Coded Decimal (BCD), BCD Numeric Format, BCD Addition
6.8 Specialized Arithmetic Hardware: Lookup ROM, Wallace Tree, Arithmetic Pipeline
6.9 Floating Point Numbers, Numeric Format
6.10 IEEE 754 Floating Point Standard, Numeric Format

7 Memory Organization **(4 hrs)**

7.1 Hierarchical Memory System
7.2 Cache Memory: Associative Memory
7.3 Cache Mapping with Associative, Direct and Set-Associative Mapping
7.4 Replacing Data in Cache, Writing Data to the Cache, Cache Performance Basics
7.5 Virtual Memory: Paging, Segmentation, and Memory Protection

8 Input /Output Organization **(6 hrs)**

8.1 Asynchronous Data Transfer
8.2 Modes of Asynchronous Data Transfer
8.3 Programmed I/O
8.4 Interrupts, Interrupts Driven Data Transfer. Types of Interrupts, Interrupts Processing, Interrupt Hardware and Priority
8.5 Direct Memory Access (DMA), DMA Transfer Modes,I/O Processors
8.6 Serial Communication, UART
8.7 USB Standards

9 Introduction to RISC **(3 hrs)**

9.1 RISC Fundamentals, RISC Instruction Set
9.2 Instruction Pipeline, Register Windows and Renaming
9.3 Conflicts in Instruction Pipeline: Data Conflicts, Branch Conflicts
9.4 RISC vs. CISC

10 Introduction to Parallel Processing **(3 hrs)**

10.1 Parallelism in Uniprocessor System

10.2 Organization of Multi-Processor System: Flynn's Taxonomy, System Topologies, MIMD System Architectures

10.3 Communication in Multi-Processor Systems: Fixed Connections and Reconfigurable Connections

10.4 Memory Organization in Multi-processor System: Shared Memory, Cache Coherence

Laboratory

Develop a project or a case study report in the field of computer Organization. The faculty concerned will provide the topic of the project work. Anoral presentationwith a demonstration in case of project should be part of work with submission of report as a component for evaluation.

Few topics of case study could be:

1. 8085/8086 Instruction Set Architecture
2. Internal Architecture of 8085/8086 Microprocessors
3. Micro-coded CPU in a Pentium Processor
4. Cache hierarchy in Itanium Processor
5. Addressing Modes in Power PC Processor
6. Parallel Processing abilities of Dual Core and Quad Core Processor
7. Advanced Features of Atom Processor
8. Systolic Arrays
9. Neural Networks

Text Book:

Carpineili, John D., *Computer System Organization and Architecture*, Addison Wesley. Pearson Education Asia (LPE.), 2001

Reference Books:

1. Hayes, John P., McGraw-Hill, Third Edition, 1998
2. W.Stalling, and Architecture, Prentice Hall India Limited. New Delhi.
3. Tanebaum, A.S., *Structured Computer Organization*, Prentice Hall India Limited, New Delhi, Fourth Edition, 1999

Database Management System (3-1-3)

Evaluation:

	Theory	Practical	Total
Sessional	30	20	50
Final	50	-	50
Total	80	20	100

Objectives:

The objective of this course is to provide fundamental concept, theory and practices in design and implementation of DBMS.

Course Contents:

1. **Introduction** **(4 hrs)**
   1.1 Concept and applications
   1.2 Objectives and Evolution
   1.3 Needs of DBMS
   1.4 Data abstraction
   1.5 Data independence
   1.6 Schema and Instances
   1.7 Concept of DDL, DML and DCL
   1.8 Database Manager and users

2. **Data Models** **(4hrs)**
   2.1 Logical, Physical and Conceptual Model
   2.2 E-R Model
   2.3 Relation with UML class diagrams
   2.4 2.4Alternate data models (Network Data Model, hierarchical Data Model)

3. **Relational Model** **(4 hrs)**
   3.1 Definitions and terminology
   3.2 Structure of relational databases
   3.3 The relational algebra
   3.4 Schema and Views
   3.5 Data dictionary

4. **Relational Database Query languages** **(8 hrs)**
   4.1 SQL – features of SQL, queries and sub-queries, Join operations, set operations and other SQL constructs
   4.2 DDL and DML queries in SQL
   4.3 Stored procedures
   4.4 QBE

5. **Database Constraints and Relational Database Design** **(8 hrs)**
   5.1 Introduction
   5.2 Integrity constraints
   5.3 Referential Integrity
   5.4 Assertions and Triggers

5.5 Functional dependencies
5.6 Normalization and Normal Forms (1NF, 2NF, 3NF, BCNF, 4NF)
5.7 Multivalued Dependencies
5.8 Decomposition of relation schemes

6. Security **(3 hrs)**
6.1 Needs of security
6.2 Security and integrity violations
6.3 Access control
6.4 Authorization
6.5 Security and Views
6.6 Encryption and decryption

7. Query Processing **(3 hrs)**
7.1 Introduction to query processing
7.2 Equivalence of expressions
7.3 Query cost estimation
7.4 Query Optimization

8. File organization and indexing **(4 hrs)**
8.1 Disks and storage
8.2 Organization of records into blocks
8.3 File organizations - The sequential and the indexed sequential file organizations
8.4 B+ Tree index
8.5 Hash index

9. Crash Recovery **(3 hrs)**
9.1 Failure classification
9.2 Concept of log-based recovery and shadow paging
9.3 Data Backup/Recovery
9.4 Remote backup system

10. Transaction Processing and Concurrency Control **(4 hrs)**
10.1 Introduction to Transactions
10.2 ACID properties of transaction
10.3 Schedules and Serializability
10.4 Concepts of locking for concurrency control

11. Advanced Database concepts **(3 hrs)**
11.1 Object-Oriented Model
11.2 Object-Relational Model (ORM)
11.3 Distributed databases
11.4 Concepts of Data Warehouses

Laboratory:
There shall be enough laboratory exercises based on some RDBMS (like ORACLE, MS-SQL server, MySQL, etc) to complement theoretical part studied. An individual project should be given to each student. 10% of sessional marks should be allocated for evaluation for lab works and project.

Text Book:

H. F. Korth and A. Silberschatz, *Database System Concepts*, McGraw Hill.

Reference Books:

1. K. Majumdar and P. Bhattacharaya, *Database Management Systems*, Tata McGraw Hill, India.
2. R. E. Mani and S. C. Nevathe, *Fundamentals of Database Systems*, Benjamin/Cummings Publishing Co. Inc.
3. G.C Everest, *Database Management*, McGraw Hill.

Numerical Methods (3-1-3)

Evaluation:

	Theory	Practical	Total
Internal	30	20	50
Final	50	-	50
Total	80	20	100

Course Objectives:

1. To introduce numerical methods for interpolation, regressions, and root finding to the solution of problems.
2. To solve elementary matrix arithmetic problems analytically and numerically.
3. To find the solution of ordinary and partial differential equations.
4. To provide knowledge of relevant high level programming language for computing, implementing, solving, and testing of algorithms.

Course Contents:

1. Solution of Nonlinear Equations **(10 hrs)**

1.1 Review of calculus and Taylor's theorem

1.2 Errors in numerical calculations

1.3 Bracketing methods for locating a root, initial approximation and convergence criteria

1.4 False position method, secant method and their convergence, Newton's method and fixed point iteration and their convergence.

2. Interpolation and Approximation **(7 hrs)**

2.1 Lagrangian's polynomials

2.2 Newton's interpolation using difference and divided differences

2.3 Cubic spline interpolation

2.4 Curve fitting: least squares lines for linear and nonlinear data

3. Numerical Differentiation and Integration **(5 hrs)**

3.1 Newton's differentiation formulas

3.2 Newton-Cote's, Quadrature formulas

3.3 Trapezoidal and Simpson's Rules

3.4 Gaussian integration algorithm

3.5 Romberg integration formulas.

4. Solution of Linear Algebraic Equations **(10 hrs)**

4.1 Matrices and their properties

4.2 Elimination methods, Gauss Jordan method, pivoting

4.3 Method of factorization: Dolittle, Crout's and Cholesky's methods

4.4 The inverse of a matrix

4.5 Ill-Conditioned systems

4.6 Iterative methods: Gauss Jacobi, Gauss Seidel, Relaxation methods

4.7 Power method.

पाठ्यक्रम विकास केन्द्र

5. **Solution of Ordinary Differential Equations** **(8 hrs)**
   5.1 Overview of initial and boundary value problems
   5.2 The Taylor's series method
   5.3 The Euler Method and its modifications
   5.4 Huen'smethod
   5.5 Runge-Kuttamethods
   5.6 Solution of higher order equations
   5.7 Boundary Value problems: Shooting method.

6. **Solution of Partial Differential Equations** **(5 hrs)**
   6.1 Review of partial differential equations
   6.2 Elliptical equations, parabolic equations, hyperbolic equations and their relevantexamples.

Laboratory:

Use of Matlab/Math-CAD/C/C++or any other relevant high level programming language for applied numerical analysis. The laboratory experiments will consist of program development and testing of:

1. Solution of nonlinear equations
2. Interpolation, extrapolation, and regression
3. Differentiation and integration
4. Linear systems of equations
5. Ordinary differential equations (ODEs)
6. Partial differential equations (PDEs)

Text Books:

1. Gerald, C. F. & Wheatly, P. O. *Applied Numerical Analysis* (7th edition).New York: Addison Wesley Publidhing Company.
2. Guha, S. & Srivastava, R. *Numerical Methods: For Engineers and Scientists*. Oxford University Press.
3. Grewal, B. S. & Grewal, J. S. *Numerical Methods in Engineering & Science* (8th edition). New Delhi: Khanna publishers. 2010.
4. Balagurusamy, E. *Numerical Methods*. New Delhi: TataMcGraw Hill. 2010.

References:

1. Moin, Parviz. *Fundamentals of Engineering Numerical Analysis*. Cambridge University Press. 2001.
2. Lindfield, G. R. & Penny, J. E. T. *Numerical Methods: Using MATLAB*. Academic Press. 2012.
3. Schilling, J. & Harris, S.L. *Applied Numerical Methods for Engineers using MATLAB and C*. Thomson publishers, 2004.
4. Sastry, S. S. *Introductory Methods of Numerical Analysis* (3rd edition). New Delhi: Prentice Hall of India. 2002.
5. Rao, S. B. & Shantha, C. K. *Numerical Methods with Programs in Basic, Fortran and Pascal*. Hyderabad: Universities Press. 2000.
6. Pratap, Rudra. *Getting Started with MATLAB*. Oxford University Press. 2010.
7. Vedamurthy, V.N. & Lyengar, N. *Numerical Methods*. Noida: Vikash Publication House. 2009.

Object Oriented Design and Modeling through UML (3-1-3)

Evaluation:

	Theory	Practical	Total
Sessional	30	20	50
Final	50	-	50
Total	80	20	100

Course Objectives:

- To explain and illustrate the fundamental concepts of object orientation
- To introduce basic concepts of object-oriented analysis and design.
- To study the main features of the software development process in an object-oriented framework.
- To provide exposure to Visual Object Oriented Modeling languages, specifically UML (Unified Modelling Language).
- To develop skills on verifying, and validating a given specification presented in UML
- To develop a specification and implementation using UML from a given system requirements description.

Course Contents:

1. **Object Oriented Fundamentals** **(10 hrs)**
   1.1. Introduction
   1.2. Object Oriented Analysis and Design
   1.3. Defining Models
   1.4. Case Study
   1.5. Requirement Process
   1.6. Use Cases
   1.7. Object Oriented Development Cycle
   1.8. Overview of the Unified Modeling Language: UML Fundamentals and Notations

2. **Object Oriented Analysis** **(12 hrs)**
   2.1. Building Conceptual Model
   2.2. Adding Associations and Attributes
   2.3. Representation of System Behavior

3. **Object Oriented Design** **(14 hrs)**
   3.1. Analysis to Design
   3.2. Describing and Elaborating Use Cases
   3.3. Collaboration Diagram
   3.4. Objects and Patterns
   3.5. Determining Visibility
   3.6. Class Diagram

पाठ्यक्रम विकास केन्द्र

4. **Implementation** **(9 hrs)**
   4.1. Programming and Development Process
   4.2. Mapping Design to Code
   4.3. Creating Class Definitions from Design Class Diagrams
   4.4. Creating Methods from Collaboration Diagram
   4.5. Updating Class Definitions
   4.6. Classes in Code
   4.7. Exception and Error Handling

Laboratory Exercises:

Laboratory Exercise will include implementing all the UML diagrams and handling a object oriented design and modeling activity in a ACSE Environment. UML pattern design and modeling will be taken up with the help of *Rational Studio 2000* or any other CASE tools.

Text Book:

Larman, C., *Applying UML and Patterns*, Pearson Education Asia, ISBN: 81-7808-336-1

Reference Books:

1. Stevens, P., Pooley, R., *Using UML: Software Engineering with Objects and Components*, Addision-Wesley, 1999, ISBN: 981-2359-15-X
2. Fowler, M., Scott, K., *UML Distilled: Applying the Standard Object Modeling Language*, Addison-Wesley, 1997, ISBN: 981-4053-59-7
3. Booch, G., Jacobson, I., Rumbaugh, J., *The Unified Software Development Process*, Addison-Wesely, 1998, ISBN: 981-235-873-0
4. Booch, G., Jacobson, I., Rumbaugh, J., *The Unified Modeling Language User Guide*, Addison-Wesely, 1998, ISBN: 981-4053-31-7
5. Jacobson I., *Object-Oriented Software Engineering – A Use Case Driven Approach*, , Addison-Wesely, 1998, ISBN: 981235994X
6. Richter C., *Designing Flexible Object-Oriented Systems with UML*, Techmedia, 2000, ISBN: 81-7635-398-1
7. Booch, G., *Object-Oriented Analysis & Design*, Pearson Education Asia, 2000, ISBN: 81-7808-156-3

Project I (0-0-3)

Evaluation:

	Theory	Practical	Total
Sessional	-	100	100
Final	-	-	-
Total	-	100	100

Course Objectives:

1. To provide the practical knowledge of project undertaking by focusing on planning, requirements elicitation, design, development and implementation of a project.
2. To provide the knowledge of Programming tools currently used in the market by carrying out a project.
3. To teach students to work and solve problem in a team environment
4. To provide the knowledge to formulate project documentation and oral presentation for his/her project.

Procedures:

The project course requires students to get themselves involved in a group on a proposed task under the direct supervision of the faculty members of their respective department. The project may be selected in consultation with the industries. The project shall be software and or electronic hardware based. The project may be done on any platform. The application shall be on any relevant areas of application e.g. Scientific Applications, Information Systems, Web Applications, Games, Simulations, Hardware based applications.

The project must be started at the beginning of the semester, span through out the semester and finished by the end of that very semester. The project should be undertaken preferably by group of students who will jointly work and implement the project. The project will be assessed by a panel of examiners as appointed by head of the department. Oral examination will be conducted by internal and external examiners as appointed by the college.

Project Work Phases:

The entire project work shall be divided in to three phases and evaluation shall be done accordingly:

First Phase: The students are required to come up with a conceptual framework for their project work which must be documented in the form of a Proposal and presented in front of an examiner in a formal presentation lasting for about 10 minutes, on the date prescribed by the college. 30% of the marks shall be based on the following criteria:

Evaluation Criteria:

Task Accomplished (20%)

- Feasibility Study
- Requirements Analysis and Specification
- Project plan

- Creativity, Innovativeness and Usefulness of the Idea

Documentation (10%)
- Proposal Report
- Estimations
- Time Line

Second Phase: The students are required to show the progress of their work done so far. They must have finished the design phase including the overall system/architectural design and validation scheme. 50% of total mark shall be based on the following criteria:

Evaluation Criteria:

Task Accomplished (40%)
- System/Architectural Design
- Depth of Project work
- Progress
- Level of achievement
- Group/Team Effort
- Ability to propose solutions

Documentation (10%)
- Report organization
- Completeness and consistency of the report
- Validation Criteria
- Organization and analysis of data and results

Third Phase: All students must have finished all phases of their project work including requirements analysis, design, coding, testing on time before Final Project Presentation.

Students must come up with a visible output of the product that they have developed and perform an oral defense of their work in the presence of an external examiner (external to the department or from industries). The final presentation should be conducted on the last week of final semester term.

Evaluation (20%):
- Presentation
- Completeness and Final Output of the Project
- Viva
- Final Project Report