CMP 226 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 & 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 Hi India.
2. R. E. Mani and S. C. Nevathe, *Fundamentals of Database Systems*, Benjamin/Cummin Publishing Co. Inc.
3. G.C Everest, *Database Management*, McGraw Hill.

MTH 214 Engineering Mathematics IV (3-2-0)

Evaluation:

	Theory	Practical	Total
Internal	50	-	50
Final	50	-	50
Total	100	-	100

Course Objectives:

After completion of this course students will be able to

- to explain and apply theorems of complex variables in their required applied problems
- to apply concepts of Fourier and Z-transform in the signal processing.
- to study wave and diffusion equations in Cartesian, cylindrical, and polar coordinates.

Course Contents:

Unit I: Complex variable **12 hrs**

1.1 Review of complex numbers with their properties
1.2 De Moirves Theorem
1.3 Function of complex variables,
1.4 Conformal mappings
1.5 Analyticity , necessary condition of analyticity
1.6 Cauchy integral theorem, Cauchy integral formula, Extension form of Cauchy integ formula,
1.7 Taylor and Laurent series
1.8 Singularities, zeros, poles, complex integration, residue theorem

Unit II: Z-transform **9 hrs**

2.1 Definition, one sided and two sided z transform
2.2 Linear Time invariant system, Unit impulse function
2.3 Properties of z transform, region of convergence
2.4 Inverse Z transform by residue and partial fraction
2.5 Parseval theorem, convolution
2.6 Application (Solution of difference equation)

Unit III: Fourier Integral and Fourier Transform **7 hrs**

3.1 Fourier series in complex form
3.2 Fourier integral, Sine integral and cosine integral
3.3 Fourier transform, cosine transform, sine transform
3.4 Inverse Fourier transform, Parseval identity
3.5 Convolution theorem and its applications

Unit IV: Partial Differential Equation **14 hrs**

4.1 Definition with examples
4.2 Method of separation of variables

4.3 Derivation and solutions of Wave equations (one and two dimensional) and their applications.
4.4 Wave equation by D Alembert's method
4.5 Derivation and solution of heat equation (one and two dimensional)and their applicati
4.6 Laplacian equation [Cartesian, polar, cylindrical, spherical form(statement only)], their solutions.
4.7 Engineering applications of partial differential equation.

Unit V: Curve in space **3 hrs**

5.1 Ellipsoid, hyperboloid, Paraboloid, cylinder, cone (Standard equations, their sketch)
5.2 Tangent line and tangent plane on the space curve

Text books:

1. E. Kreyszig, *Advanced Engineering Mathematics*,8th edition Wiley-Easter Publication, New Delhi
2. H. K. Dass & R. Verma, Higher Engineering Mathematics, First edition, S. Chand & Company Limited, New Delhi

Reference Books:

1. Digital Signal Processing: J. G. Proakis, Prentice Hall of India.
2. V Sundaran, R Bala Subramanayam, K. L . Laxminarayanam, *Engineering Mathematics* Volume II
3. A. V. Oppenheim, *Discrete-Time Signal Processing*, Prentice Hall, India Limited, 1990.
4. K. Ogata, *Discrete-Time Control System*, Prentice Hall, India Limited, 1993.

CMP 214 Microprocessors and Assembly Language Programming (3-1-3)

Evaluation:

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

Course Objectives:
The purpose of the course is to provide the basics fundamentals and operations of microprocessor. It provides knowledge to program microprocessor using assembly language and design microprocessor based systems and interfaces.

Course Contents:

1. Introduction to microprocessor **(3 hrs)**
- 1.1 Brief description: Microprocessor, Microcontroller, Microcomputer
- 1.2 Application of microprocessor
- 1.3 Evolution of microprocessor: INTEL series

2. Architectural Details and Instruction set of 8085 and 8086 microprocessor (10 hrs)
- 2.1 Internal architecture and description
- 2.2 Instruction set
- 2.3 Addressing modes
- 2.4 Instruction cycle, Machine cycle, t-states
- 2.5 Timing Diagram

3. Assembly Language Programming **(12 hrs)**
- 3.1 Introduction
- 3.2 Format of an assembly language instruction
- 3.3 Basic assembly language programs of 8085
- 3.4 ALP development tools:Editor,Assembler,Linker,Debuger,Locator,Emulator
- 3.5 Macro Assembler and Assembler Directives
- 3.6 8086 Assembly Language Programs in MASM/TASM
- 3.7 Modular Programming
  - 3.7.1 Linking and Relocation
  - 3.7.2 Stacks Procedures
  - 3.7.3 Macros Program Design
  - 3.7.4 String Manipulation

4. Bus Structure and Memory Devices **(4 hrs)**
- 4.1 Introduction: Data/Address/Control bus
- 4.2 Synchronous and Asynchronous bus
- 4.3 Memory Classification
- 4.4 Memory Interfacing and Addressing Decoding

5. Interrupt **(6 hrs)**

5.1 Introduction
5.2 Interrupt Sources: Hardware, Software, Processor
5.3 Interrupt Types: Maskable, Non-Maskable Interrupt
5.4 8086 Interrupts
5.5 Interrupt Vector Table
5.6 Vector Chain and Polled Interrupt
5.7 Interrupt Processing

6. Input / Output Interfaces **(10 hrs)**

6.1 Serial I/O standards: 8251A USART
6.2 8259A Programmable Interrupt Controller(PIC)
6.3 8255A Programmable Peripheral Interface(PPI)
6.4 8254 Programmable Interrupt Timer(PIT) and its application
6.5 DMA and DMA controller

Laboratory:

1. A minimum of 10 laboratory exercises shall be done with the use of SDK-85/SDK-86 o equivalent microprocessor trainer kit and Simulators.
2. Numerous assembly language programming exercises are to be done both with the hel of microprocessor trainer kit and Macro-Assemblers in PC.

Text Books:

1. Liu. Yu-cheng and Gibson Glenn A., Microprocessor Systems: The 8080 8088 famil Architecture. Programming and Design.PHI, 1998. ISBN: 81-203-0409-8
2. Brey. Barry B..Intel Microprocessors.PHI. 1998. ISBN:

References:

1. Antonakos. J. L. An Introduction to the Intel family of microprocessors, 3rd ed, Pearso Education Asia. ISBN: 81-7808-312-4
2. Triebel, Walter A. and Singh Avvbtar, The 8088 and 8086 microprocessors Programming Interfacing, Software, Hardware, and Applications PHI. 1998, ISBN
3. L.A Leventhal, Introduction to Microprocessor software, Hardware & Programmin Prentice Hall of India. Pvt. Ltd., 1995.
4. A.P. Malvino, An Introduction to Microcomputers. Prentice Hall of India. Pvt. Ltd 1995
5. P.K. Ghosh, P.R. Sridhar, 0000 to 8085;Introduction to Microprocessor for Engineers and Scientists, Prentice Hall of India Pvt. Ltd 1997
6. Rajaraman, V. and Radhakrishnan T., Essentials of Assembly Language Programming for the IBM PC, PHI, 1998. ISBN: 81-203-1425-5

CMP 212 Programming in Java

Evaluation:

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

Course Objectives:

The purpose of the course is to provide the concept of objective oriented programming using Java. It provides sound understanding of network programming and database connectivity. Moreover, it provides front end platform for development of applications.

Course Contents:

1. **Elements Of Java Language** **(3 hrs)**
   - 1.1 Java as a Programming tools, Benefits of Java, Historical Background of Java
   - 1.2 A simple Java Program, Data type, Variable, Assignment and Initialization, Operator, String, Control Flow
   - 1.3 Class Method (User Defined Function), Array

2. **Object Oriented Programming In Java** **(9 hrs)**
   - 2.1 Introduction to object oriented programming in Java
   - 2.2 Reusability using Existing classes
   - 2.3 Building User defined class, Package
   - 2.4 Inheritance
   - 2.5 Casting Abstract classes
   - 2.6 Access Protection Mechanism
   - 2.7 Reflection
   - 2.8 Designing Inheritance
   - 2.9 Interface, Inner Classes

3. **Exception, Stream and I/O** **(3 hrs)**
   - 3.1 Handling Error and Exception, catching Exception, tips on handling Exception, Debugging techniques
   - 3.2 Stream, Zip files Stream, Object Stream
   - 3.3 Handling Files

4. **Applets and Application** **(4 hrs)**
   - 4.1 Fundamental concept of Applet, Simple Applet
   - 4.2 Testing Applets, Converting Application to Applets
   - 4.3 Applets HTML tags and Attribute. Pop –UP Windows in Applet
   - 4.4 Multimedia Applets context

5. Events, Handling Events and AWT/Swing **(6 hrs)**

5.1 Basic of Event handling, AWT Event hierarchy
5.2 Semantics and low level Events in AWT, Event Handling
5.3 Individual Events. Separating GUI and Application code
5.4 Multicasting, Advance Event Handling
5.5 An Introduction of layout management, Text input choice, scroll Bar
5.6 Complex layout management, Menus, Dialog Box

6. Graphics and Images / Animation / Multimedia **(5 hrs)**

6.1 Introduction to Graphics Programming, creating Closable frames
6.2 Terminating graphics program. Frame layout displaying information in a frame
6.3 Graphics object. Text and fonts, color
6.4 Drawing shapes from lines drawing rectangle and Ovals
6.5 Filling shapes paint mode images

7. Network Programming **(8 hrs)**

7.1 Networking Basics
7.2 Introduction to Socket
7.3 Socket Programming
7.4 Understanding Port
7.5 Networking Classes in Java
7.6 Creating Own Server and Client in Java
7.7 Creating Multithread Java Server
7.8 URL and URL connection Class

8. Java Database Connectivity (JDBC)

8.1 Understanding JDBC **(7 hrs)**
8.2 Database Driver
8.3 JDBC-ODBC bridge
8.4 Java Native Driver
8.5 Intermediate Database Access Server
8.6 JDBC API
8.7 Making a JDBC Application
8.8 Using Prepared Statement

References:

1. Dietel H.M and Dietel P.J., Java: How to Program, Third Edition, Pearson Educati Asia
2. Naughton Java 2: The Complete Reference, Tata McGraw Hill
3. Balagurusamy E., Programming in Java: 2nd Edition, Tata McGraw Hill

CMP 290 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. The must have finished the design phase including the overall system/architectural design an 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 includin 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 perfor an oral defense of their work in the presence of an external examiner (external to the departmen or from industries). The final presentation should be conducted on the last week of fina semester term.

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

CMP 220 Software Engineering Fundamentals (3-0-3)

Evaluation:

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

Course Objectives:

The purpose of the course is to introduce the concepts and techniques required to direct and control the development of medium to large-scale software, including project management, quality assurance, software process improvement and software metrics. It aims to broaden student's understanding of possible software development paradigms (e. g., structured analysis and design, object-oriented approaches). Moreover, it enables to explore some of the problems of software maintenance.

Course Contents:

1. Software Project Management Concepts **(2 hrs)**

1.1 Software: Crisis and Myths, Software Process and Process Models

1.2 Process technology, Product and Process. People, Product, Process, Project

2. Software Metrics **(3 hrs)**

2.1 Measures, Metrics, and Indicators: Software Measurement

2.2 Metrics for software quality, Statistical Quality Control

2.3 Metrics for Small Organizations

3. Software Project Planning and Risk **(3 hrs)**

3.1 Objectives, Scope, Resources, Project Estimation, Decomposition Techniques

3.2 Empirical Estimation Models, Risk Management Strategies

3.3 Software Risks, Risk Identification, Risk Projection

4. Software Quality Assurance **(5 hrs)**

4.1 Concepts, Software Quality Assurance

4.2 Software Reviews, Formal Technical Reviews

4.3 Formal Approaches to SQA

4.4 Statistical Quality Assurance. Software Reliability

4.5 ISO 9000 Quality Standards, SQA Plan

5. Software Configuration Management **(4 hrs)**

5.1 Software Configuration Management, SCM Process

5.2 Identification of Objects in the Software Configuration, Version Control
5.3 Change Control, Configuration Audit
5.4 Status Reporting, SCM Standards

6. Analysis Concepts and Principles **(5 hrs)**

6.1 Requirements Analysis, Analysis Principles
6.2 Software Prototyping. Specification and Specification Review
6.3 Analysis Modeling: Elements of Analysis Model
6.4 Data Modeling Functional Modeling and Information Flow
6.5 Behavioral Modeling, Structured Analysis- Data Dictionary

7. Design Concepts and Principles **(6 hrs)**

7.1 Design Process, Principles and Concepts
7.2 Architectural and Component Level Design
7.3 Software Architecture, Data Design, Architectural Styles
7.4 Mapping Requirements into a Software Architecture
7.5 Transform Mapping, Transaction Mapping
7.6 Structured Programming, Comparison of Design Notation

8. Software Testing Techniques and Strategies

8.1 Testing Fundamentals, **(7 hrs)**
8.2 Test Case Design.
8.3 White Box Testing. Basis Path Testing,
8.4 Control Structure Testing.
8.5 Black-Box Testing. Unit Testing,
8.6 Integration Testing,
8.7 Validation Testing, System Testing

9. Object-Oriented Concepts and Principles

9.1 Object-Oriented Paradigm **(4 hrs)**
9.2 Object-Oriented Concepts
9.3 Identifying the Elements of an Object Model
9.4 Management of Object-Oriented Software Projects

10. Object-Oriented Analysis and Design

10.1 Domain Analysis **(6 hrs)**
10.2 Components of the OO Analysis Model
10.3 The OOA Process, Design for Object-Oriented Systems
10.4 The System Design Process
10.5 The Object Design Process
10.6 Design Patterns

Laboratory:

The Laboratory Exercise includes System Analysis, Design, Development, and Testing. Debugging of a small Real Life problem and then attempting to visualize various Software Engineering activities, like Revision Control System, Version Management, Library Building, etc. using some of the Software Engineering Tool or CASE Tool.

Reference Books:

1. Mall. R., Foundations of Software Engineering. PH1. 2000. ISBN: 81-203-1445-
2. Pressman. R. S., Software Engineering a practitioners Approach. 5th Edition. McGraw Hill. 2001. ISBN: 0-07-118458-9
3. Somnterville. I., Software Engineering, 5th Edition. Addison -Wesley. 1995. ISBN: 0-201-43579 -9