# MANAGEMENT AND ENTREPRENEURSHIP FOR IT INDUSTRY

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)

**SEMESTER – V**

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Exam Marks</th>
<th>Total Number of Lecture Hours</th>
<th>Exam Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>15CS51</td>
<td>20</td>
<td>80</td>
<td>50</td>
<td>03</td>
</tr>
</tbody>
</table>

**CREDITS – 04**

**Course objectives:** This course will enable students to

- Explain the principles of management, organization and entrepreneur.
- Discuss on planning, staffing, ERP and their importance
- Infer the importance of intellectual property rights and relate the institutional support

<table>
<thead>
<tr>
<th>Module – 1</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Introduction</strong> - Meaning, nature and characteristics of management, scope and Functional areas of management, goals of management, levels of management, brief overview of evolution of management theories. Planning- Nature, importance, types of plans, steps in planning, Organizing- nature and purpose, types of Organization, Staffing- meaning, process of recruitment and selection</td>
<td>10 Hours</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module – 2</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Directing and controlling</strong>- meaning and nature of directing, leadership styles, motivation Theories, Communication- Meaning and importance, Coordination- meaning and importance, Controlling- meaning, steps in controlling, methods of establishing control.</td>
<td>10 Hours</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module – 3</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Entrepreneur</strong> – meaning of entrepreneur, characteristics of entrepreneurs, classification and types of entrepreneurs, various stages in entrepreneurial process, role of entrepreneurs in economic development, entrepreneurship in India and barriers to entrepreneurship. Identification of business opportunities, market feasibility study, technical feasibility study, financial feasibility study and social feasibility study.</td>
<td>10 Hours</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module – 4</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Preparation of project and ERP</strong> - meaning of project, project identification, project selection, project report, need and significance of project report, contents, formulation, guidelines by planning commission for project report, <strong>Enterprise Resource Planning: Meaning and Importance- ERP</strong> and Functional areas of Management – Marketing / Sales- Supply Chain Management – Finance and Accounting – Human Resources – Types of reports and methods of report generation</td>
<td>10 Hours</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module – 5</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Micro and Small Enterprises:</strong> Definition of micro and small enterprises, characteristics and advantages of micro and small enterprises, steps in establishing micro and small enterprises, Government of India industrial policy 2007 on micro and small enterprises, case study (Microsoft), Case study(Captain G R Gopinath),case study (N R Narayana Murthy &amp; Infosys), <strong>Institutional support:</strong> MSME-DI, NSIC, SIDBI, KIADB, KSSIDC, TECSON, KSFC, DIC and District level single window agency, <strong>Introduction to IPR.</strong></td>
<td>10 Hours</td>
</tr>
</tbody>
</table>

**Course outcomes:** The students should be able to:

- Define management, organization, entrepreneur, planning, staffing, ERP and outline
their importance in entrepreneurship
- Utilize the resources available effectively through ERP
- Make use of IPRs and institutional support in entrepreneurship

**Question paper pattern:**
The question paper will have TEN questions.
There will be TWO questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer FIVE full questions, selecting ONE full question from each module.

**Text Books:**

**Reference Books:**
2. Entrepreneurship Development - S S Khanka - S Chand & Co.
# COMPUTER NETWORKS

[As per Choice Based Credit System (CBCS) scheme]

(Effective from the academic year 2016 -2017)

## SEMESTER – V

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Number of Lecture Hours/Week</th>
<th>Exam Marks</th>
<th>Total Number of Lecture Hours</th>
<th>Exam Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>15CS52</td>
<td>20</td>
<td>4</td>
<td>80</td>
<td>50</td>
<td>03</td>
</tr>
</tbody>
</table>

**CREDITS – 04**

**Course objectives:** This course will enable students to

- Demonstration of application layer protocols
- Discuss transport layer services and understand UDP and TCP protocols
- Explain routers, IP and Routing Algorithms in network layer
- Disseminate the Wireless and Mobile Networks covering IEEE 802.11 Standard
- Illustrate concepts of Multimedia Networking, Security and Network Management

## Module – 1

### **Application Layer:**


**T1: Chap 2**

### **Module – 2**

### **Transport Layer:**


**T1: Chap 3**

### **Module – 3**

### **The Network layer:**


### Module – 4


**T1: Chap: 4 : 4.3-4.7**

---

### Module – 5


**Network Support for Multimedia:** Dimensioning Best-Effort Networks, Providing Multiple Classes of Service, Diffserv, Per-Connection Quality-of-Service (QoS) Guarantees: Resource Reservation and Call Admission

**T1: Chap: 6 : 6.4-6.8**

---

### Course outcomes:

The students should be able to:

- Explain principles of application layer protocols
- Recognize transport layer services and infer UDP and TCP protocols
- Classify routers, IP and Routing Algorithms in network layer
- Understand the Wireless and Mobile Networks covering IEEE 802.11 Standard
- Describe Multimedia Networking and Network Management

### Question paper pattern:

The question paper will have TEN questions.

There will be TWO questions from each module.

Each question will have questions covering all the topics under a module.

The students will have to answer FIVE full questions, selecting ONE full question from each module.

### Text Books:


### Reference Books:

2. Larry L Peterson and Brusce S Davie, Computer Networks, fifth edition, ELSEVIER
3. Andrew S Tanenbaum, Computer Networks, fifth edition, Pearson
## DATABASE MANAGEMENT SYSTEM

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)

**SEMESTER – V**

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Exam Marks</th>
<th>Total Number of Lecture Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>15CS53</td>
<td></td>
<td>80</td>
<td>50</td>
</tr>
</tbody>
</table>

**CREDITS – 04**

**Course objectives:** This course will enable students to

- Provide a strong foundation in database concepts, technology, and practice.
- Practice SQL programming through a variety of database problems.
- Demonstrate the use of concurrency and transactions in database
- Design and build database applications for real world problems.

<table>
<thead>
<tr>
<th>Module – 1</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction to Databases: Introduction, Characteristics of database approach, Advantages of using the DBMS approach, History of database applications. Overview of Database Languages and Architectures: Data Models, Schemas, and Instances. Three schema architecture and data independence, database languages, and interfaces, The Database System environment. Conceptual Data Modelling using Entities and Relationships: Entity types, Entity sets, attributes, roles, and structural constraints, Weak entity types, ER diagrams, examples, Specialization and Generalization. Textbook 1: Ch 1.1 to 1.8, 2.1 to 2.6, 3.1 to 3.10</td>
<td>10 Hours</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module – 2</th>
<th>10 Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relational Model: Relational Model Concepts, Relational Model Constraints and relational database schemas, Update operations, transactions, and dealing with constraint violations. Relational Algebra: Unary and Binary relational operations, additional relational operations (aggregate, grouping, etc.) Examples of Queries in relational algebra. Mapping Conceptual Design into a Logical Design: Relational Database Design using ER-to-Relational mapping. SQL: SQL data definition and data types, specifying constraints in SQL, retrieval queries in SQL, INSERT, DELETE, and UPDATE statements in SQL, Additional features of SQL. Textbook 1: Ch4.1 to 4.5, 5.1 to 5.3, 6.1 to 6.5, 8.1; Textbook 2: 3.5</td>
<td></td>
</tr>
</tbody>
</table>

| Module – 3 | 10 Hours |
| SQL : Advances Queries: More complex SQL retrieval queries, Specifying constraints as assertions and action triggers, Views in SQL, Schema change statements in SQL. Database Application Development: Accessing databases from applications, An introduction to JDBC, JDBC classes and interfaces, SQLJ, Stored procedures, Case study: The internet Bookshop. Internet Applications: The three-Tier application architecture, The presentation layer, The Middle Tier Textbook 1: Ch7.1 to 7.4; Textbook 2: 6.1 to 6.6, 7.5 to 7.7. | |

<p>| Module – 4 | 10 Hours |</p>
<table>
<thead>
<tr>
<th>Form. <strong>Normalization Algorithms:</strong> Inference Rules, Equivalence, and Minimal Cover, Properties of Relational Decompositions, Algorithms for Relational Database Schema Design, Nulls, Dangling tuples, and alternate Relational Designs, Further discussion of Multivalued dependencies and 4NF, Other dependencies and Normal Forms</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Textbook 1:</strong> Ch14.1 to 14.7, 15.1 to 15.6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Module – 5</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Transaction Processing:</strong> Introduction to Transaction Processing, Transaction and System concepts, Desirable properties of Transactions, Characterizing schedules based on recoverability, Characterizing schedules based on Serializability, Transaction support in SQL. <strong>Concurrency Control in Databases:</strong> Two-phase locking techniques for Concurrency control, Concurrency control based on Timestamp ordering, Multiversion Concurrency control techniques, Validation Concurrency control techniques, Granularity of Data items and Multiple Granularity Locking. <strong>Introduction to Database Recovery Protocols:</strong> Recovery Concepts, NO-UNDO/REDO recovery based on Deferred update, Recovery techniques based on immediate update, Shadow paging, Database backup and recovery from catastrophic failures</td>
</tr>
<tr>
<td><strong>Textbook 1:</strong> 20.1 to 20.6, 21.1 to 21.7, 22.1 to 22.4, 22.7.</td>
</tr>
</tbody>
</table>

| **10 Hours** |

<table>
<thead>
<tr>
<th><strong>Course outcomes:</strong> The students should be able to:</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Identify, analyze and define database objects, enforce integrity constraints on a database using RDBMS.</td>
</tr>
<tr>
<td>• Use Structured Query Language (SQL) for database manipulation.</td>
</tr>
<tr>
<td>• Design and build simple database systems</td>
</tr>
<tr>
<td>• Develop application to interact with databases.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Question paper pattern:</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>The question paper will have TEN questions.</td>
</tr>
<tr>
<td>There will be TWO questions from each module.</td>
</tr>
<tr>
<td>Each question will have questions covering all the topics under a module.</td>
</tr>
<tr>
<td>The students will have to answer FIVE full questions, selecting ONE full question from each module.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Text Books:</strong></th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th><strong>Reference Books:</strong></th>
</tr>
</thead>
</table>
AUTOMATA THEORY AND COMPUTABILITY
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)

SEMESTER – V

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Exam Marks</th>
</tr>
</thead>
<tbody>
<tr>
<td>15CS54</td>
<td>20</td>
<td>80</td>
</tr>
</tbody>
</table>

CREDITS – 04

Course objectives: This course will enable students to

- Introduce core concepts in Automata and Theory of Computation
- Identify different Formal language Classes and their Relationships
- Design Grammars and Recognizers for different formal languages
- Prove or disprove theorems in automata theory using their properties
- Determine the decidability and intractability of Computational problems

Module – 1 Teaching Hours

Why study the Theory of Computation, Languages and Strings: Strings, Languages. A Language Hierarchy, Computation, Finite State Machines (FSM): Deterministic FSM, Regular languages, Designing FSM, Nondeterministic FSMs, From FSMs to Operational Systems, Simulators for FSMs, Minimizing FSMs, Canonical form of Regular languages, Finite State Transducers, Bidirectional Transducers.

Textbook 1: Ch 1,2, 3,4, 5.1 to 5.10

10 Hours

Module – 2

Regular Expressions (RE): what is a RE?, Kleene’s theorem, Applications of REs, Manipulating and Simplifying REs. Regular Grammars: Definition, Regular Grammars and Regular languages. Regular Languages (RL) and Non-regular Languages: How many RLs, To show that a language is regular, Closure properties of RLs, to show some languages are not RLs.

Textbook 1: Ch 6, 7, 8: 6.1 to 6.4, 7.1, 7.2, 8.1 to 8.4

10 Hours

Module – 3

Context-Free Grammars(CFG): Introduction to Rewrite Systems and Grammars, CFGs and languages, designing CFGs, simplifying CFGs, proving that a Grammar is correct, Derivation and Parse trees, Ambiguity, Normal Forms. Pushdown Automata (PDA): Definition of non-deterministic PDA, Deterministic and Non-deterministic PDAs, Non-determinism and Halting, alternative equivalent definitions of a PDA, alternatives that are not equivalent to PDA.

Textbook 1: Ch 11, 12: 11.1 to 11.8, 12.1, 12.2, 12.4, 12.5, 12.6

10 Hours

Module – 4


10 Hours

Module – 5

Variants of Turing Machines (TM), The model of Linear Bounded automata: Decidability: Definition of an algorithm, decidability, decidable languages,
Undecidable languages, halting problem of TM, Post correspondence problem. Complexity: Growth rate of functions, the classes of P and NP, Quantum Computation: quantum computers, Church-Turing thesis.  

**Textbook 2:** Ch 9.7 to 9.8, 10.1 to 10.7, 12.1, 12.2, 12.8, 12.8.1, 12.8.2

**Course outcomes:** The students should be able to:

- Acquire fundamental understanding of the core concepts in automata theory and Theory of Computation
- Learn how to translate between different models of Computation (e.g., Deterministic and Non-deterministic and Software models).
- Design Grammars and Automata (recognizers) for different language classes and become knowledgeable about restricted models of Computation (Regular, Context Free) and their relative powers.
- Develop skills in formal reasoning and reduction of a problem to a formal model, with an emphasis on semantic precision and conciseness.
- Classify a problem with respect to different models of Computation.

**Question paper pattern:**
The question paper will have TEN questions. There will be TWO questions from each module. Each question will have questions covering all the topics under a module. The students will have to answer FIVE full questions, selecting ONE full question from each module.

**Text Books:**


**Reference Books:**

5. Basavaraj S. Anami, Karibasappa K G, Formal Languages and Automata theory, Wiley India, 2012
**OBJECT ORIENTED MODELING AND DESIGN**

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)

**SEMESTER – V**

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Total Number of Lecture Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>15CS551</td>
<td>20</td>
<td>40</td>
</tr>
</tbody>
</table>

**CREDITS – 03**

**Course objectives:** This course will enable students to

- Describe the concepts involved in Object-Oriented modelling and their benefits.
- Demonstrate concept of use-case model, sequence model and state chart model for a given problem.
- Explain the facets of the unified process approach to design and build a Software system.
- Translate the requirements into implementation for Object Oriented design.
- Choose an appropriate design pattern to facilitate development procedure.

### Module – 1

**Introduction, Modelling Concepts and Class Modelling:** What is Object orientation? What is OO development? OO Themes; Evidence for usefulness of OO development; OO modelling history. Modelling as Design technique: Modelling; abstraction; The Three models. Class Modelling: Object and Class Concept; Link and associations concepts; Generalization and Inheritance; A sample class model; Navigation of class models; Advanced Class Modelling, Advanced object and class concepts; Association ends; N-ary associations; Aggregation; Abstract classes; Multiple inheritance; Metadata; Reification; Constraints; Derived Data; Packages.

**Text Book-1: Ch 1, 2, 3 and 4**

<table>
<thead>
<tr>
<th>Teaching Hours</th>
<th>8 Hours</th>
</tr>
</thead>
</table>

### Module – 2

**UseCase Modelling and Detailed Requirements:** Overview; Detailed object-oriented Requirements definitions; System Processes-A use case/Scenario view; Identifying Input and outputs-The System sequence diagram; Identifying Object Behaviour-The state chart Diagram; Integrated Object-oriented Models.

**Text Book-2: Chapter- 6:Page 210 to 250**

<table>
<thead>
<tr>
<th>Teaching Hours</th>
<th>8 Hours</th>
</tr>
</thead>
</table>

### Module – 3

**Process Overview, System Conception and Domain Analysis:** Process Overview: Development stages; Development life Cycle; System Conception: Devising a system concept; elaborating a concept; preparing a problem statement. Domain Analysis: Overview of analysis; Domain Class model: Domain state model; Domain interaction model; Iterating the analysis.

**Text Book-1: Chapter- 10,11, and 12**

<table>
<thead>
<tr>
<th>Teaching Hours</th>
<th>8 Hours</th>
</tr>
</thead>
</table>

### Module – 4

**Use case Realization :The Design Discipline within up iterations:** Object Oriented Design-The Bridge between Requirements and Implementation; Design Classes and Design within Class Diagrams; Interaction Diagrams-Realizing Use Case and defining methods; Designing with Communication Diagrams; Updating the Design Class Diagram; Package Diagrams-Structuring the Major Components; Implementation Issues for Three-Layer Design.

**Text Book-2: Chapter 8: page 292 to 346**

<table>
<thead>
<tr>
<th>Teaching Hours</th>
<th>8 Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module – 5</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td></td>
</tr>
<tr>
<td><strong>Design Patterns:</strong> Introduction; what is a design pattern?, Describing design patterns, the catalogue of design patterns, Organizing the catalogue, How design patterns solve design problems, how to select a design patterns, how to use a design pattern; Creational patterns: prototype and singleton (only); structural patterns adaptor and proxy (only). <strong>Text Book-3: Ch-1: 1.1, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, Ch-3, Ch-4.</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Course outcomes:</strong> The students should be able to:</td>
<td></td>
</tr>
<tr>
<td>• Describe the concepts of object-oriented and basic class modelling.</td>
<td></td>
</tr>
<tr>
<td>• Draw class diagrams, sequence diagrams and interaction diagrams to solve problems.</td>
<td></td>
</tr>
<tr>
<td>• Choose and apply a befitting design pattern for the given problem.</td>
<td></td>
</tr>
<tr>
<td><strong>Question paper pattern:</strong></td>
<td></td>
</tr>
<tr>
<td>The question paper will have TEN questions.</td>
<td></td>
</tr>
<tr>
<td>There will be TWO questions from each module.</td>
<td></td>
</tr>
<tr>
<td>Each question will have questions covering all the topics under a module.</td>
<td></td>
</tr>
<tr>
<td>The students will have to answer FIVE full questions, selecting ONE full question from each module.</td>
<td></td>
</tr>
<tr>
<td><strong>Text Books:</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Reference Books:</strong></td>
<td></td>
</tr>
</tbody>
</table>
INTRODUCTION TO SOFTWARE TESTING
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)

SEMESTER – V

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Lecture Hours/Week</td>
<td>3</td>
<td>Exam Marks</td>
</tr>
<tr>
<td>Total Number of Lecture Hours</td>
<td>40</td>
<td>Exam Hours</td>
</tr>
</tbody>
</table>

CREDITS – 03

Course objectives: This course will enable students to

- Differentiate the various testing techniques.
- Analyze the problem and derive suitable test cases.
- Apply suitable technique for designing of flow graph.
- Explain the need for planning and monitoring a process.

<table>
<thead>
<tr>
<th>Module – 1</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basics of Software Testing: Basic definitions, Software Quality, Requirements, Behaviour and Correctness, Correctness versus Reliability, Testing and Debugging, Test cases, Insights from a Venn diagram, Identifying test cases, Test-generation Strategies, Test Metrics, Error and fault taxonomies, Levels of testing, Testing and Verification, Static Testing. Textbook 3: Ch 1:1.2 - 1.5, 3; Textbook 1: Ch 1</td>
<td>8 Hours</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module – 2</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem Statements: Generalized pseudo code, the triangle problem, the NextDate function, the commission problem, the SATM (Simple Automatic Teller Machine) problem, the currency converter, Saturn windshield wiper Functional Testing: Boundary value analysis, Robustness testing, Worst-case testing, Robust Worst testing for triangle problem, NextDate problem and commission problem, Equivalence classes, Equivalence test cases for the triangle problem, NextDate function, and the commission problem, Guidelines and observations, Decision tables, Test cases for the triangle problem, NextDate function, and the commission problem, Guidelines and observations. Textbook 1: Ch 2, 5, 6 &amp; 7, Textbook 2: Ch 3</td>
<td>8 Hours</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module – 3</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Fault Based Testing: Overview, Assumptions in fault based testing, Mutation analysis, Fault-based adequacy criteria, Variations on mutation analysis. Structural Testing: Overview, Statement testing, Branch testing, Condition testing, Path testing: DD paths, Test coverage metrics, Basis path testing, guidelines and observations, Data –Flow testing: Definition-Use testing, Slice-based testing, Guidelines and observations. T2:Chapter 16, 12 T1:Chapter 9 &amp; 10</td>
<td>8 Hours</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module – 4</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Execution: Overview of test execution, from test case specification to test cases, Scaffolding, Generic versus specific scaffolding, Test oracles, Self-checks as oracles, Capture and replay Process Framework: Basic principles: Sensitivity, redundancy, restriction, partition, visibility, Feedback, the quality process, Planning and monitoring, Quality goals, Dependability properties, Analysis Testing, Improving the process, Organizational factors. Planning and Monitoring the Process: Quality and process, Test and analysis strategies and plans, Risk planning, monitoring the process, Improving the</td>
<td>8 Hours</td>
</tr>
</tbody>
</table>
Module – 5

**Integration and Component-Based Software Testing:** Overview, Integration testing strategies, Testing components and assemblies. System, Acceptance and Regression Testing: Overview, System testing, Acceptance testing, Usability, Regression testing, Regression test selection techniques, Test case prioritization and selective execution. **Levels of Testing, Integration Testing:** Traditional view of testing levels, Alternative life-cycle models, The SATM system, Separating integration and system testing, A closer look at the SATM system, Decomposition-based, call graph-based, Path-based integrations.

*T2: Chapter 21 & 22, T1: Chapter 12 & 13*

<table>
<thead>
<tr>
<th>Course outcomes: The students should be able to:</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Derive test cases for any given problem</td>
</tr>
<tr>
<td>• Compare the different testing techniques</td>
</tr>
<tr>
<td>• Classify the problem into suitable testing model</td>
</tr>
<tr>
<td>• Apply the appropriate technique for the design of flow graph.</td>
</tr>
<tr>
<td>• Create appropriate document for the software artefact.</td>
</tr>
</tbody>
</table>

**Question paper pattern:**
The question paper will have TEN questions. There will be TWO questions from each module. Each question will have questions covering all the topics under a module. The students will have to answer FIVE full questions, selecting ONE full question from each module.

**Text Books:**

**Reference Books:**
# Advanced Java and J2EE

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)

**SEMESTER – V**

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Exam Marks</th>
<th>Total Number of Lecture Hours</th>
<th>Exam Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>15CS553</td>
<td>20</td>
<td>80</td>
<td>40</td>
<td>3</td>
</tr>
</tbody>
</table>

**CREDITS – 03**

**Course objectives:** This course will enable students to
- Identify the need for advanced Java concepts like Enumerations and Collections
- Construct client-server applications using Java socket API
- Make use of JDBC to access database through Java Programs
- Adapt servlets to build server side programs
- Demonstrate the use of JavaBeans to develop component-based Java software

## Module – 1

**Enumerations, Autoboxing and Annotations(metadata):** Enumerations, Enumeration fundamentals, the values() and valueOf() Methods, java enumerations are class types, enumerations Inherits Enum, example, type wrappers, Autoboxing, Autoboxing and Methods, Autoboxing/Unboxing occurs in Expressions, Autoboxing/Unboxing, Boolean and character values, Autoboxing/Unboxing helps prevent errors, A word of Warning. Annotations, Annotation basics, specifying retention policy, Obtaining Annotations at run time by use of reflection, Annotated element Interface, Using Default values, Marker Annotations, Single Member annotations, Built-In annotations.

### Teaching Hours

<table>
<thead>
<tr>
<th>Module – 1</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enumerations, Autoboxing and Annotations</td>
<td>8 Hours</td>
</tr>
</tbody>
</table>

## Module – 2


### Teaching Hours

<table>
<thead>
<tr>
<th>Module – 2</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>The collections and Framework</td>
<td>8 Hours</td>
</tr>
</tbody>
</table>

## Module – 3

**String Handling :** The String Constructors, String Length, Special String Operations, String Literals, String Concatenation, String Concatenation with Other Data Types, String Conversion and toString( ) Character Extraction, charAt( ), getChars( ), getBytes( ) toCharArray(), String Comparison, equals( ) and equalsIgnoreCase( ), regionMatches( ) startsWith( ) and endsWith( ), equals( ) Versus == , compareTo( ) Searching Strings, Modifying a String, substring( ), concat( ), replace( ), trim( ), Data Conversion Using valueOf( ), Changing the Case of Characters Within a String, Additional String Methods, StringBuffer, StringBuffer Constructors, length( ) and capacity( ), ensureCapacity( ), setLength( ), charAt( ) and setCharAt( ), getChars( ), append( ), insert( ), reverse( ), delete( ) and deleteCharAt( ), replace( ), substring( ), Additional StringBuffer Methods, StringBuilder

### Text Book 1: Ch 15

### Teaching Hours

<table>
<thead>
<tr>
<th>Module – 3</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>String Handling</td>
<td>8 Hours</td>
</tr>
</tbody>
</table>
### Module – 4

| Background; The Life Cycle of a Servlet; Using Tomcat for Servlet Development; A simple Servlet; The Servlet API; The Javax.servlet Package; Reading Servlet Parameter; The Javax.servlet.http package; Handling HTTP Requests and Responses; Using Cookies; Session Tracking. Java Server Pages (JSP): JSP, JSP Tags, Tomcat, Request String, User Sessions, Cookies, Session Objects |
|---|---|
| **Text Book 1:** Ch 31 | **Text Book 2:** Ch 11 |

**8 Hours**

### Module – 5

| The Concept of JDBC; JDBC Driver Types; JDBC Packages; A Brief Overview of the JDBC process; Database Connection; Associating the JDBC/ODBC Bridge with the Database; Statement Objects; ResultSet; Transaction Processing; Metadata, Data types; Exceptions. |
|---|---|
| **Text Book 2:** Ch 06 |

**8 Hours**

### Course outcomes:
The students should be able to:
- Interpret the need for advanced Java concepts like enumerations and collections in developing modular and efficient programs
- Build client-server applications and TCP/IP socket programs
- Illustrate database access and details for managing information using the JDBC API
- Describe how servlets fit into Java-based web application architecture
- Develop reusable software components using Java Beans

### Question paper pattern:
The question paper will have TEN questions. There will be TWO questions from each module. Each question will have questions covering all the topics under a module. The students will have to answer FIVE full questions, selecting ONE full question from each module.

### Text Books:

### Reference Books:
ADVANCED ALGORITHMS  
[As per Choice Based Credit System (CBCS) scheme]  
(Effective from the academic year 2016-2017)  

SEMESTER – V  

Subject Code 15CS554  
IA Marks 20  
Number of Lecture Hours/Week 3  
Exam Marks 80  
Total Number of Lecture Hours 40  
Exam Hours 03  

CREDITS – 03  

Course objectives: This course will enable students to  
• Explain principles of algorithms analysis approaches  
• Compare and contrast a number theoretic based strategies.  
• Describe complex signals and data flow in networks  
• Apply the computational geometry criteria.  

Module – 1  

Teaching Hours 8 Hours  

Module – 2  

8 Hours  

Module – 3  
DFT and FFT efficient implementation of FFT, Graph Algorithms, Bellman-Ford Algorithm Shortest paths in a DAG, Johnson's Algorithm for sparse graphs, Flow networks and the Ford-Fulkerson Algorithm, Maximum bipartite matching.  

8 Hours  

Module – 4  
Computational Geometry-I: Geometric data structures using, C, Vectors, Points, Polygons, Edges Geometric objects in space; Finding the intersection of a line and a triangle, Finding star-shaped polygons using incremental insertion.  

8 Hours  

Module – 5  
Computational Geometry-II: Clipping: Cyrus-Beck and Sutherland-Hodman Algorithms; Triangulating, monotonic polygons; Convex hulls, Gift wrapping and Graham Scan; Removing hidden surfaces  

8 Hours  

Course outcomes: The students should be able to:  
• Explain the principles of algorithms analysis approaches  
• Apply different theoretic based strategies to solve problems  
• Illustrate the complex signals and data flow in networks with usage of tools  
• Describe the computational geometry criteria.  

Question paper pattern:  
The question paper will have TEN questions.  
There will be TWO questions from each module.  
Each question will have questions covering all the topics under a module.  
The students will have to answer FIVE full questions, selecting ONE full question from each
module.

<table>
<thead>
<tr>
<th>Text Books:</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Reference Books:</th>
</tr>
</thead>
</table>
# COMPUTER NETWORK LABORATORY

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)

**SEMESTER – V**

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Number of Lecture Hours/Week</th>
<th>Exam Marks</th>
<th>Total Number of Lecture Hours</th>
<th>Exam Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>15CSL57</td>
<td>20</td>
<td>01L + 02P</td>
<td>80</td>
<td>40</td>
<td>03</td>
</tr>
</tbody>
</table>

**CREDITS – 02**

**Course objectives:** This course will enable students to

- Demonstrate operation of network and its management commands
- Simulate and demonstrate the performance of GSM and CDMA
- Implement data link layer and transport layer protocols.

**Description (If any):**

For the experiments below modify the topology and parameters set for the experiment and take multiple rounds of reading and analyze the results available in log files. Plot necessary graphs and conclude. Use NS2/NS3.

**Lab Experiments:**

**PART A**

1. Implement three nodes point – to – point network with duplex links between them. Set the queue size, vary the bandwidth and find the number of packets dropped.
2. Implement transmission of ping messages/trace route over a network topology consisting of 6 nodes and find the number of packets dropped due to congestion.
3. Implement an Ethernet LAN using n nodes and set multiple traffic nodes and plot congestion window for different source / destination.
4. Implement simple ESS and with transmitting nodes in wire-less LAN by simulation and determine the performance with respect to transmission of packets.
5. Implement and study the performance of GSM on NS2/NS3 (Using MAC layer) or equivalent environment.
6. Implement and study the performance of CDMA on NS2/NS3 (Using stack called Call net) or equivalent environment.

**PART B**

**Implement the following in Java:**

7. Write a program for error detecting code using CRC-CCITT (16- bits).
8. Write a program to find the shortest path between vertices using bellman-ford algorithm.
9. Using TCP/IP sockets, write a client – server program to make the client send the file name and to make the server send back the contents of the requested file if present.
10. Write a program on datagram socket for client/server to display the messages on client side, typed at the server side.
11. Write a program for simple RSA algorithm to encrypt and decrypt the data.
12. Write a program for congestion control using leaky bucket algorithm.

**Study Experiment / Project:**

NIL

**Course outcomes:** The students should be able to:

- Analyze and Compare various networking protocols.
- Demonstrate the working of different concepts of networking.
- Implement, analyze and evaluate networking protocols in NS2 / NS3

<table>
<thead>
<tr>
<th>Conduction of Practical Examination:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. All laboratory experiments are to be included for practical examination.</td>
</tr>
<tr>
<td>2. Students are allowed to pick one experiment from part A and part B with lot.</td>
</tr>
<tr>
<td>3. Strictly follow the instructions as printed on the cover page of answer script</td>
</tr>
<tr>
<td>Part A: 10+25+5 =40</td>
</tr>
<tr>
<td>Part B: 10+25+5 =40</td>
</tr>
<tr>
<td>5. Change of experiment is allowed only once and marks allotted to the procedure part to be made zero.</td>
</tr>
</tbody>
</table>
DBMS LABORATORY WITH MINI PROJECT
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – V

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Number of Lecture Hours/Week</th>
<th>Exam Marks</th>
<th>Total Number of Lecture Hours</th>
<th>Exam Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>15CSL58</td>
<td>20</td>
<td>01L + 02P</td>
<td>80</td>
<td>40</td>
<td>03</td>
</tr>
</tbody>
</table>

CREDITS – 02

Course objectives: This course will enable students to

- Foundation knowledge in database concepts, technology and practice to groom students into well-informed database application developers.
- Strong practice in SQL programming through a variety of database problems.
- Develop database applications using front-end tools and back-end DBMS.

Description (If any):

PART-A: SQL Programming (Max. Exam Mks. 50)
- Design, develop, and implement the specified queries for the following problems using Oracle, MySQL, MS SQL Server, or any other DBMS under LINUX/Windows environment.
- Create Schema and insert at least 5 records for each table. Add appropriate database constraints.

PART-B: Mini Project (Max. Exam Mks. 30)
- Use Java, C#, PHP, Python, or any other similar front-end tool. All applications must be demonstrated on desktop/laptop as a stand-alone or web based application (Mobile apps on Android/IOS are not permitted.)

Lab Experiments:

Part A: SQL Programming

1. Consider the following schema for a Library Database:
   BOOK(Book_id, Title, Publisher_Name, Pub_Year)
   BOOK_AUTHORS(Book_id, Author_Name)
   PUBLISHER(Name, Address, Phone)
   BOOK_COPIES(Book_id, Branch_id, No-of_Copies)
   BOOK_LENDING(Book_id, Branch_id, Card_No, Date_Out, Due_Date)
   LIBRARY_BRANCH(Branch_id, Branch_Name, Address)
   Write SQL queries to
   1. Retrieve details of all books in the library – id, title, name of publisher, authors, number of copies in each branch, etc.
   2. Get the particulars of borrowers who have borrowed more than 3 books, but from Jan 2017 to Jun 2017.
   3. Delete a book in BOOK table. Update the contents of other tables to reflect this data manipulation operation.
   4. Partition the BOOK table based on year of publication. Demonstrate its working with a simple query.
   5. Create a view of all books and its number of copies that are currently available in the Library.

2. Consider the following schema for Order Database:
   SALESMAN(Salesman_id, Name, City, Commission)
   CUSTOMER(Customer_id, Cust_Name, City, Grade, Salesman_id)
   ORDERS(Ord_No, Purchase_Amt, Ord_Date, Customer_id, Salesman_id)
   Write SQL queries to
   1. Count the customers with grades above Bangalore’s average.
2. Find the name and numbers of all salesman who had more than one customer.
3. List all the salesman and indicate those who have and don't have customers in their cities (Use UNION operation.)
4. Create a view that finds the salesman who has the customer with the highest order of a day.
5. Demonstrate the DELETE operation by removing salesman with id 1000. All his orders must also be deleted.

Consider the schema for Movie Database:
ACTOR(Act_id, Act_Name, Act_Gender)
DIRECTOR(Dir_id, Dir_Name, Dir_Phone)
MOVIES(Mov_id, Mov_Tittle, Mov_Year, Mov_Lang, Dir_id)
MOVIE_CAST(Act_id, Mov_id, Role)
RATING(Mov_id, Rev_Stars)

Write SQL queries to
1. List the titles of all movies directed by ‘Hitchcock’.
2. Find the movie names where one or more actors acted in two or more movies.
3. List all actors who acted in a movie before 2000 and also in a movie after 2015 (use JOIN operation).
4. Find the title of movies and number of stars for each movie that has at least one rating and find the highest number of stars that movie received. Sort the result by movie title.
5. Update rating of all movies directed by ‘Steven Spielberg’ to 5.

Consider the schema for College Database:
STUDENT(USN, SName, Address, Phone, Gender)
SEMSEC(SSID, Sem, Sec)
CLASS(USN, SSID)
SUBJECT(Subcode, Title, Sem, Credits)
IAMARKS(USN, Subcode, SSID, Test1, Test2, Test3, FinalIA)

Write SQL queries to
1. List all the student details studying in fourth semester ‘C’ section.
2. Compute the total number of male and female students in each semester and in each section.
3. Create a view of Test1 marks of student USN ‘1BI15CS101’ in all subjects.
4. Calculate the FinalIA (average of best two test marks) and update the corresponding table for all students.
5. Categorize students based on the following criterion:
   If FinalIA = 17 to 20 then CAT = ‘Outstanding’
   If FinalIA = 12 to 16 then CAT = ‘Average’
   If FinalIA < 12 then CAT = ‘Weak’
   Give these details only for 8th semester A, B, and C section students.

Consider the schema for Company Database:
EMPLOYEE(SSN, Name, Address, Sex, Salary, SuperSSN, DNo)
DEPARTMENT(DNo, DName, MgrSSN, MgrStartDate)
DLOCATION(DNo, DLoc)
PROJECT(PNo, PName, PLocation, DNo)
WORKS_ON(SSN, PNo, Hours)

Write SQL queries to
1. Make a list of all project numbers for projects that involve an employee whose last name is ‘Scott’, either as a worker or as a manager of the department that controls the project.
2. Show the resulting salaries if every employee working on the ‘IoT’ project is given a 10 percent raise.
3. Find the sum of the salaries of all employees of the ‘Accounts’ department, as well as the maximum salary, the minimum salary, and the average salary in this department.
4. Retrieve the name of each employee who works on all the projects controlled by department number 5 (use NOT EXISTS operator).
5. For each department that has more than five employees, retrieve the department number and the number of its employees who are making more than Rs. 6,00,000.

**Part B: Mini project**

- For any problem selected, write the ER Diagram, apply ER-mapping rules, normalize the relations, and follow the application development process.
- Make sure that the application should have five or more tables, at least one trigger and one stored procedure, using suitable frontend tool.
- Indicative areas include; health care, education, industry, transport, supply chain, etc.

**Course outcomes:** The students should be able to:

- Create, Update and query on the database.
- Demonstrate the working of different concepts of DBMS
- Implement, analyze and evaluate the project developed for an application.

**Conduction of Practical Examination:**

1. All laboratory experiments from part A are to be included for practical examination.
2. Mini project has to be evaluated for 30 Marks.
3. Report should be prepared in a standard format prescribed for project work.
4. Students are allowed to pick one experiment from the lot.
5. Strictly follow the instructions as printed on the cover page of answer script.
6. Marks distribution:
   a) Part A: Procedure + Conduction + Viva:10 + 35 +5 =50 Marks
   b) Part B: Demonstration + Report + Viva voce = 15+10+05 = 30 Marks
7. Change of experiment is allowed only once and marks allotted to the procedure part to be made zero.