University of Pune
Revised Structure for the B. Sc. (Computer Science) Course
(Second Year to be implemented from Academic Year 2014-2015)

S. Y. B. Sc. (Computer Science)

No	Paper	Title: Semester I	Title: Semester II
1	Computer Science Paper I	Data Structures using 'C'	Object Oriented Concepts using C++
2	Computer Science Paper II	Relational Database Management System	Software Engineering
3	Computer Science Paper III	Data structures Practicals and C++ Practicals	
4	Mathematics Paper I	Mathematics Paper I- Sem I	Mathematics Paper I- Sem II
5	Mathematics Paper II	Mathematics Paper II-Sem I	Mathematics Paper II- Sem II
6	Mathematics Paper III	Practical Course in Mathematics	
7	Electronics Paper I	Electronics Paper I-Sem I	Electronics Paper I-Sem II
8	Electronics Paper II	Electronics Paper II-Sem I	Electronics Paper II-Sem II
9	Electronics Paper III	Practical Course in Electronics	
10	English	Technical English- Sem I	Technical English – Sem II
11	Computer Science Paper IV	Database Practicals & Mini Project using Software Engineering techniques	

University of Pune
Proposed draft Syllabus for the S. Y. B. Sc. (Computer Science) Course
(From Academic Year 2014-2015)

Subject : Computer Science

No	Paper	Title: Semester I	Title: Semester II
1	Computer Science Paper I	CS 211- Data Structures using 'C'	CS 221- Object Oriented Concepts using C++
2	Computer Science Paper II	CS-212 - Relational Database Management System (RDBMS)	CS 222-Software Engineering
3	Computer Science Paper III	Data structures Practicals and C++ Practicals	
11	Computer Science Paper IV	Database Practicals & Mini Project using Software Engineering techniques	

Teaching Scheme :- For Theory courses :4 Lectures Per Week, Total Lectures in each semester 48; For Practical courses : 4 Lectures per Week per Batch of 15 students, Total Lectures 100 per year.

Examination Scheme :- For Theory Courses : There will be 40 : 10 pattern of evaluation in which 40 marks are for Semester End examination to be conducted at the end of each semester and 10 marks for semester wise internal evaluation.

For practical courses :Practical examination will be conducted at the end of academic year, 80 marks will be assigned to performance in practical examination and 20 marks for journals through continuous assessment.

In case of Paper IV, marks out of 100 will be converted to grades

Marks	Grade
75 and above	O
65 and above	A
55 and above	B
50 and above	C
45 and above	D
40 and above	E
Below 40 (indicates Failure)	F

Computer Science Theory Paper I
SEM II
CS 211- DATA STRUCTURES USING C
(Compulsory Course)

Total Lectures: 48

Objective:

1. To learn the systematic way of solving problem
2. To understand the different methods of organizing large amount of data
3. To efficiently implement the different data structures
4. To efficiently implement solutions for specific problems

Prerequisites: Knowledge of C Programming Language

1. Introduction to data structures [2]

1.1 Concept
1.2 Data type, Data object, ADT
1.3 Need of Data Structure
1.4 Types of Data Structure

2. Algorithm analysis [2]

2.1 Algorithm – definition, characteristics
2.2 Space complexity, time complexity
2.3 Asymptotic notation (Big O, Omega Ω)

3. Linear data structures [6]

3.1 Introduction to Arrays - array representation
3.2 Sorting algorithms with efficiency
- bubble sort, Insertion sort, Merge sort, Quick Sort

4. Linked List [6]

4.1 Introduction to List
4.2 Implementation of List – static & dynamic representation,
4.3 Types of Linked List
4.4 Operations on List
4.5 Applications of Linked List – polynomial manipulation
4.6 Generalized linked list – concept & representation

5. Stacks [6]

5.1 Introduction
5.2 Representation-static & dynamic
5.3 Operations
5.4 Application - infix to postfix & prefix, postfix evaluation,
5.5 Simulating recursion using stack

6. Queues [6]

6.1 Introduction

6.2 Representation -static & dynamic
6.3 Operations
6.4 Circular queue, priority queue (with implementation)
6.5 Concept of doubly ended queue

7. Trees **[12]**
7.1 Concept & Terminologies
7.2 Binary tree, binary search tree
7.3 Representation – static & dynamic
7.4 Operations on BST – create, Insert, delete, traversals (preorder, inorder, postorder), counting leaf, non-leaf & total nodes , non recursive inorder traversal
7.5 Application - Heap sort
7.6 Height balanced tree- AVL trees- Rotations

8. Graph **[8]**
8.1 Concept & terminologies
8.2 Graph Representation – Adjacency matrix, adjacency list, inverse adjacency list, adjacency multilist, orthogonal list
8.3 Traversals – BFS & DFS
8.4 Applications – AOV network – topological sort , AOE network – critical path , Dijkstra's Shortest path algorithm

References:

1. Fundamentals of Data Structures ---- By Horowitz Sahani (Galgotia)
2. Data Structures using C and C++ --- By Yedidyah Langsam, Aaron M. Tenenbaum, Moshe J. Augenstein
3. Introduction to Data Structures using C---By Ashok Kamthane
4. Data Structures using C --- Bandopadhyay & Dey (Pearson)

Computer Science Theory Paper I
SEM II
CS 221 -Object Oriented Concepts and Programming in C++
(Compulsory Course)

Total Lectures: 48

Objective:-

Acquire an understanding of basic object oriented concepts and the issues involved in effective class design

Write C++ programs that use object oriented concepts such as information hiding, constructors, destructors, inheritance etc.

Prerequisites: Knowledge of C Programming Language

1. Object oriented concepts [2]

1.1 Object oriented methodology
1.2 Features, advantages and Applications of OOPS

2. Introduction to C++ [8]

2.1 Data types, new operators and keywords, type conversion in C++
2.2 Introduction to reference variables
2.3 Classes & Objects
2.4 Access specifiers
2.5 Defining data members and member functions
2.6 Array of objects
2.7 Managing console I/O
2.8 C++ stream classes
2.9 Formatted and unformatted console I/O
2.10 Usage of manipulators

3. Function in C++ [8]

3.1 Call by reference, Return by reference
3.2 Function overloading and default arguments
3.3 Inline function
3.4 Static class members
3.5 Friend functions

4. Constructors and destructor [4]

4.1 Types of constructors
4.2 memory allocation (new and delete)
4.3 Destructor

5. Operator overloading [4]

5.1 overloading unary and binary operators
5.2 overloading using friend function
5.3 usage of this pointer
5.4 overloading insertion and extraction operator

6. Inheritance [10]

6.1 types of inheritance with examples
6.2 constructor and destructor in derived class
6.3 virtual base classes
6.4 abstract base classes , virtual functions and pure virtual function

7. Working with files [6]

7.1 File operations
7.2 File pointer and their manipulation
7.3 File updation with random access

8. Templates [4]

8.1 Introduction to templates
8.2 Class templates, function templates and overloading of function templates
8.3 Templates with multiple parameters
8.4 CASE study on STL (with reference to container classes, operational utilities)

9. Exception Handling in C++ [2]

9.1 try, catch and throw primitives

Reference Books: -

1. Object Oriented Programming with C++ by Robert Lafore
2. Object Oriented Programming with C++ by E. Balagurusamy
3. Object Oriented Modeling and Design by James Rumbough
4. The Complete Reference C++ by Herbert Schildt
5. Let us C++ by – Yashwant Kanitkar

Computer Science Theory Paper – II
SEM I
CS-212-Relational Database Management System (RDBMS)
(Compulsory Course)

Total Lectures: 48

Objective:-

- To teach fundamental concepts of RDBMS (PL/PgSQL)
- To teach principles of databases
- To teach database management operations
- To teach data security and its importance
- To teach client server architecture

Prerequisites: Knowledge of DBMS

1. **Relational Database Design** **[14]**

1.1 Preliminaries

Functional Dependencies
Basic concepts
Closure of a set of functional dependencies,
Closure of attribute set,
Canonical cover
Decomposition

1.2 4th Normal form, Multivalued Dependencies,
1.3 PL/PgSqL: Datatypes, Language structure
1.4 Stored Functions
1.5 Controlling the program flow, conditional statements, loops
1.6 Handling errors and exceptions
1.7 Cursors, Stored Procedures, Views and Triggers

2 **Transaction Concepts and concurrency control** **[14]**

2.10 Describe a transaction, properties of transaction, state of the transaction.
2.11 Executing transactions concurrently associated problem in concurrent execution.
2.12 Schedules, types of schedules, concept of Serializability, precedence graph for Serializability.
2.13 Ensuring Serializability by locks, different lock modes, 2PL and its variations.
2.14 Basic timestamp method for concurrency, Thomas Write Rule.
2.15 Locks with multiple granularity, dynamic database concurrency (Phantom Problem).
2.16 Timestamps versus locking.
2.17 Deadlock handling methods
2.18 Detection and Recovery (Wait for graph).
2.19 Prevention algorithms (Wound-wait, Wait-die)

3 **Database Integrity and Security Concepts** **[8]**

3.10 Domain constraints
3.11 Referential Integrity
3.12 Introduction to database security concepts
3.13 Methods for database security
3.14 Discretionary access control method
3.15 Mandatory access control and role base access control for multilevel security.
3.16 Use of views in security enforcement.
3.17 Overview of encryption technique for security.
3.18 Statistical db security.

4 **Crash Recovery** **[8]**

4.1 Failure classification
4.2 Recovery concepts
4.3 Log base recovery techniques (Deferred and Immediate update)
4.4 Checkpoints
4.5 Recovery with concurrent transactions (Rollback, checkpoints, commit)

4.6 Database backup and recovery from catastrophic failure.
4.7 Shadow paging

5. **Client-Server Technology** **[4]**
5.1 Describe client-server computing.
5.2 Evolution of Client - Server information systems.
5.3 Client – Server Architecture benefits.
5.4 Client Server Architecture
- Components, Principles, Client Components
- Communication middleware components
- Database middleware components
- Client Server Databases

References:-

1. Fundamentals of Database Systems (4th Ed) By: Elmasri and Navathe
2. Database System Concepts (4th Ed) By: Korth, Sudarshan, Silberschatz
3. Practical PostgreSQL O’REILLY

Computer Science Theory Paper – II
SEM II
CS 222- Software Engineering
(Compulsory Course)

Total Lectures: 48
Objective:-

- To teach basics of System Analysis and Design.
- To teach principles of Software Engineering
- To teach various process models used in practice
- To know about the system engineering and requirement engineering
- To build analysis model

Prerequisites: Basic knowledge of DBMS

1. Introduction to System Analysis and Design **[8]**

1.1 System definition and concepts: Characteristics and types of system, Manual and automated systems

1.2 Real-life Business sub-systems: Production, Marketing, Personal, Material, Finance

1.3 Systems models types of models: Systems environment and boundaries, Real-time and distributed systems, Basic principles of successful systems

1.4 System Analyst
Role and need of systems analyst ,Qualifications and responsibilities , Systems Analyst as and agent of change,

1.3 System Development cycle
Introduction to systems development life cycle (SDLC) :
Various phases of development : Analysis, Design, Development, Implementation, Maintenance

1.4 Systems documentation considerations: Principles of systems documentation , Types of documentation and their importance, Enforcing documentation discipline in an organization .

2.Introduction To Software Engineering **[4]**

2.1 The Evolving Role of Software
2.2 Software
2.3 The Changing Nature of Software
2.4 Legacy Software
2.5. The Quality of Legacy Software
2.6. Software Evolution
27 Software Myths

3. Process and Process Models **[6]**

3.1 Software Engineering – A Layered Technology
3.2 A Process Framework
3.4 Process Technology
3.5 Product and Process
3. 6 Process Models
3.7 Prescriptive Models
3.8The Waterfall Model
3.9 Incremental Process Models
3.9.1 The Incremental Model
3.9.2 The RAD Model
3.10 Evolutionary Process Models
3.10.1 Prototyping
3.10.2 The Spiral Model
3.10.3 The Concurrent Development Model
3.10.4 A Final Comment of Evolutionary Processes

4. An Agile View of Process **[4]**

4.1 What Is Agility?
4.2 What Is an Agile Process?
4.2.1 The Politics of Agile Development
4.2.2 Human Factors
4.3 Agile Process Models
4.3.1 Extreme Programming (XP)
4.3.2 Adaptive Software Development (ASD)
4.3.3 Dynamic Systems Development Method (DSDM)
4.3.4 Scrum
4.3.5 Crystal
4.3.6 Feature Driven Development (FDD)
4.3.7 Agile Modeling (AM)

5. Software Engineering Practice **[4]**

Software Engineering Practice
The Essence of Practice
Core Principles
Communication Practices
Planning Practices
Modeling Practices
Analysis Modeling Principles
Design Modeling Principles

6. System Engineering **[4]**

6.1 Computer-Based Systems
6.2 The System Engineering Hierarchy
6.2.1 System Modeling
6.2.2 System Simulation
6.3 Business Process Engineering: An Overview

7. Requirements Engineering **[10]**

7.1 A Bridge to Design and Construction
7.2 Requirements Engineering Tasks
7.2.1 Inception
7.2.2 Elicitation
7.2.3 Elaboration
7.2.4 Negotiation
7.2.5 Specification
7.2.6 Validation
7.2.7 Requirements Management
7.3 Initiating the Requirements Engineering Process
7.3.1 Identifying the Stakeholders
7.3.2 Recognizing Multiple Viewpoints
7.3.3 Working Toward Collaboration
7.3.4 Asking the First Questions
7.4 Eliciting Requirements
4.4.1 Collaborative Requirements Gathering
4.4.2 Quality Function Deployment
4.4.3 User Scenarios
4.4.4 Elicitation Work Products
7.5 Building the Analysis Model
7.5.1 Elements of the Analysis Model
7.5.2 Analysis Patterns
7.6 Negotiating Requirements
7.7 Validating Requirements

8. Building the Analysis Model **[8]**

8.1 Requirements Analysis
8.1.1 Overall objective and Philosophy
8.1.2 Analysis rule of Thumb
8.1.3 Domain Analysis
8.2 Analysis Modeling Approaches
8.3 Data Modeling Concepts
8.3.1 Data Objects
8.3.2 Data Attributes
8.3.3 Relationships
8.3.4 Cardinality and Modality

Reference Books:

1. Software Engineering – A Practitioner’s Approach 7th Edition – Roger S. Pressman [McGraw Hill International Edition]

2. System analysis and design – Elias M.Awad.

3. Software Engineering – IAN Sommerville 7th / 8th Edition (Pearson Edition)

4.System analysis and design –Perry Edwards

5. Analysis and design of information systems – James A.Senn

Important to Note: It is absolutely necessary and essential that all the practicals for Paper III and Paper IV be conducted on Free and Open Source Operating System like Linux. All the practicals related to C and C++ needs to be conducted using GCC compiler. Database Systems to be used is PostGreSQL

Paper III – Computer Science Practical Paper I

Title : Data Structures using C and C++ Assignments

Objective :-

1. Design and implement Data structures and related algorithms
2. Understand several ways of solving the same problem.

Syllabus

Computer Science : Paper III : Data Structures using C and C++ Assignments		
No	Topic	Lectures
1	Sorting Algorithms – Bubble sort, Insertion	4

2	Recursive Sorting Algorithms – Quick sort , Merge Sort	4
3	Static/Dynamic stack implementation, infix to postfix, infix to prefix and evaluation of Postfix.	8
4	Static and Dynamic Queue Implementation – Circular queue, Priority queue	8
5	Singly Linked List, Doubly Linked List and Circular Linked List.	8
6	Polynomial addition (Using Linked list).	4
7	Binary Search Tree Traversal: Create, add, delete, display nodes.	8
8	Adjacency matrix to adjacency list conversion, in degree, out degree,	4
9	Graph: DFS, BFS.	4
10	Shortest path Dijkstra algorithm.	4
11	Class , Object and methods implementation ,	4
12	Constructor: Copy Constructor, Default Constructor, Parameterized Constructor	4
13	Memory Allocation: new and delete operators , dynamic constructor	4
14	Inline function, friend function, default argument,	4
15	Function Overloading.	4
16	Operator overloading.	8
17	Inheritance: Single, multiple, multilevel, hierarchy.	8
18	File Handling: Updation of files using random access exists clauses	4
19	Exception Handling	4

Paper IV – Computer Science Practical Paper II

Title : Database Assignments and Mini Project using Software Engineering techniques
Objective :-

i) Understanding the use of cursors, triggers, views and stored procedures
ii) Understanding the steps of system analysis and design
iii) Understanding Data requirements for a specific problem domain
iv) Designing Data base as per the Data requirements
v) Designing queries as per the functional requirements

Syllabus

Computer Science : Paper IV : Database Assignments and Mini Project for Designing Backend

using Software Engineering techniques practicals		
No	Topic	Lectures
1	Simple Queries	4
2	Nested Queries, using aggregate functions	4
3	Queries using cursors	8
4	Queries using loops and conditional statements	8
5	Exceptional handling	4
6	Stored procedures	8
7	Triggers and views	12
8	Problem definition , scope	8
9	Feasibility study	4
10	Gathering Data Requirements and Functional Requirement	12
11	ERD	4
12	Designing the Database	8
13	Designing queries related to Functional requirements	12
14	Designing stored procedures	8