Savitribai Phule Pune University

(Formerly University of Pune)

Three Year B.Sc. Degree Program in Computer Science

(Faculty of Science & Technology)

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

Choice Based Credit System Syllabus

To be implemented from Academic Year 2019-2020

Title of the Course: B. Sc. (Computer Science)

Preamble:

The B. Sc. (Computer Science) course is systematically designed three year degree program under the faculty of Science and Technology. The objective of the course is to prepare students to undertake careers involving problem solving using computer science and technologies, or to pursue advanced studies and research in computer science. The syllabus which comprises of Computer Science subject along with that of the three allied subjects (Mathematics, Electronics and Statistics) covers the foundational aspects of computing sciences and also develops the requisite professional skills and problem solving abilities using computing sciences.

Introduction:
At the first year of under-graduation, the basic foundations of two important skills required for software development are laid. A course in problem solving and programming along with a course in database fundamentals forms the preliminary skill set for solving computational problems. The practical courses are designed to supplement the theoretical training in the year. Along with Computer Science, the two theoretical and one practical course each in Statistics, Mathematics and Electronics help in building a strong foundation. Career Advancement courses are introduced in both semesters to cover additional areas of Computer Science.

At the second year of under-graduation, computational problem solving skills are further strengthened by a course in Data structures. Software engineering concepts that are required for project design are also introduced. Essential concepts of computer networking are also introduced in this year. The practical course included in both semesters complements the theory courses.

At the third year of under-graduation, all the subjects are designed to fulfill core Computer Science requirements as well as meet the needs of the software industry. Theory courses are adequately supplemented by hands-on practical courses. Skill Enhancement courses enable the students to acquire additional value-added skills.

Objectives:

- To develop problem solving abilities using a computer.
- To build the necessary skill set and analytical abilities for developing computer based solutions for real life problems.
- To train students in professional skills related to Software Industry.
- To prepare necessary knowledge base for research and development in Computer Science.
- To help students build-up a successful career in Computer Science and to produce entrepreneurs who can innovate and develop software products.

Titles of Papers, Credit Allocation and Scheme of Evaluation

Semester I (Total credits=22)

Course type	Paper Code	Paper title	Credits		Evaluation		
			T	P	CA	UA	TOTAL
CC-I	CS-111	Problem Solving using Computer and 'C' Programming	2		15	35	50
	CS-112	Database Management Systems	2		15	35	50
	CS-113	Practical course based on CS101 and CS102		1.5	15	35	50
CC-II*		Mathematics – I, II and III					
CC-III*		Electronics – I,II and III					
CC-IV*		Statistics – I, II and III					

Semester II (Total credits=22)

Course type	Paper Code	Paper title	Credits		Evaluation		
			T	P	CA	UA	TOTAL
CC-V	CS-121	Advanced 'C' Programming	2		15	35	50
	CS-122	Relational Database Management Systems	2		15	35	50
	CS-123	Practical course based on CS201 and CS202		1.5	15	35	50
CC-VI*		Mathematics – I,II and III					
CC-VII*		Electronics – I, II and III					
CC-VIII*		Statistics – I,II and III					

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

Semester III (Total credits=22)

Course type	Paper Code	Paper title	Credits		Evaluation		
			T	P	CA	UA	TOTAL
CC-IX	CS-231	Data Structures and Algorithms – I	2		15	35	50
	CS-232	Software Engineering	2		15	35	50
	CS-233	Practical course based on CS301		2	15	35	50
CC-X*		Mathematics – I, II and III					
CC-XI*		Electronics – I,II and III					
AECC-I*		Environment Science – I	2				
AECC-II*		Language Communication – I	2				

Semester IV (Total credits=22)

Course type	Paper Code	Paper title	Credits		Evaluation		
			T	P	CA	UA	TOTAL
CC-XII	CS-241	Data Structures and Algorithms – II	2		15	35	50
	CS-242	Computer Networks - I	2		15	35	50
	CS-243	Practical course based on CS401		2	15	35	50
CC-XIII*		Mathematics – I,II and III					
CC-XIV*		Electronics – I, II and III					
AECC-III*		Environment Science – I	2				
AECC-IV*		Language Communication – I	2				

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

Semester V (Total credits=22)

Course type	Paper Code	Paper title	Credits		Evaluation		
			T	P	CA	UA	TOTAL
DSEC - I	CS-351	Operating Systems - I	2		15	35	50
	CS-352	Computer Networks - II	2		15	35	50
	CS-357	Practical course based on CS501		2	15	35	50
DSEC - II	CS-353	Web Technologies - I	2				
	CS-354	Foundations of Data Science	2				
	CS-358	Practical course based on CS503		2			
DSEC - III	CS-355	Object Oriented Programming - I (Core Java)	2				
	CS-356	Theoretical Computer Science and Compiler Construction - I	2				
	CS-359	Practical Course based on CS505		2			
SECC - I	CS-3510	Python Programming / R Programming	1	1	15	35	50
SECC - II	CS-3511	Open Elective	1	1	15	35	50

Semester VI (Total credits=22)

Course type	Paper Code	Paper title	Credits		Evaluation		
			T	P	CA	UA	TOTAL
DSEC - IV	CS-361	Operating Systems - II	2		15	35	50
	CS-362	Software Testing	2		15	35	50
	CS-367	Practical course based on CS601		2	15	35	50
DSEC - V	CS-363	Web Technologies - II	2				
	CS-364	Data Analytics	2				
	CS-368	Practical course based on CS603 and CS604		2			
DSEC - VI	CS-365	Object Oriented Programming - II (Advanced Java)	2				
	CS-366	Theoretical Computer Science and Compiler Construction - II	2				
	CS-369	Practical Course based on CS605		2			
SECC- III	CS-3610	Mobile Application Development OR Software Testing Tools	1	1	15	35	50
SECC - IV	CS-3611	Project OR Open Elective	1	1	15	35	50

Detailed Syllabus:

Semester- I
Paper - I

Course Type: Core Credit **Course Code: CS101**

Course Title : Problem Solving Using Computer and 'C' Programming - I

Teaching Scheme	No. of Credits	Examination Scheme
2 Hours / Week	2	IE : 15 Marks UE: 35 Marks

Course Objectives

1. To introduce the foundations of computing, programming and problem- solving using computers.
2. To develop the ability to analyze a problem and devise an algorithm to solve it.
3. To formulate algorithms, pseudocodes and flowcharts for arithmetic and logical problems
4. To understand structured programming approach.
5. To develop the basic concepts and terminology of programming in general.
6. To implement algorithms in the 'C' language.
7. To test, debug and execute programs.

Course Outcomes:- On completion of this course, students will be able to :

1. Explore algorithmic approaches to problem solving.
2. Develop modular programs using control structures and arrays in 'C'.

Course Contents

Chapter 1	**Problem Solving Aspects**	**5 Hours**

1.1. Introduction to problem solving using computers.
1.2. Problem solving steps.
1.3 Algorithms-definition, characteristics , examples ,advantages and limitations.
1.4 Flowcharts - definition, notations , examples , advantages and limitations, Comparison with algorithms.
1.5 Pseudo codes - notations, examples, advantages and limitations.
1.6 Programming Languages as tools, programming paradigms, types of languages
1.7 Converting pseudo-code to programs.
1.8 Compilation process (compilers , interpreters), linking and loading, syntax and semantic errors, testing a program
1.9 Good Programming Practices (naming conventions , documentation, indentation).

Chapter 2	**'C' Fundamentals**	**7 Hours**

2.1 History of 'C' language.
2.2 Application areas.
2.2 Structure of a 'C' program.
2.3 'C' Program development life cycle.

2.4 Function as building blocks.
2.5 'C' tokens
2.6 Character set, Keywords , Identifiers
2.7 Variables, Constants (character, integer, float, string, escape sequences, enumeration constant).
2.8 Data Types (Built-in and user defined data types).
2.9 Operators, Expressions, types of operators, Operator precedence and Order of evaluation.
2.10 Character input and output.
2.11 String input and output.
2.12 Formatted input and output.

Chapter 3	**Control Structures**	**6 Hours**

3.1 Decision making structures:- if ,if-else, switch and conditional operator.
3.2 Loop control structures:- while ,do while, for.
3.3 Use of break and continue.
3.4 Nested structures.
3.5 Unconditional branching (goto statement).

Chapter 4	**Functions**	**6 Hours**

4.1 Concept of function, Advantages of Modular design.
4.2 Standard library functions.
4.3 User defined functions:- declaration , definition, function call, parameter passing (by value), return statement.
4.4 Recursive functions.
4.5 Scope of variables and Storage classes.

Chapter 5	**Arrays**	**6 Hours**

5.1 Concept of array.
5.2 Types of Arrays – One , Two and Multidimensional array.
5.3 Array Operations - declaration, initialization, accessing array elements.
5.4 Memory representation of two-dimensional array (row major and column major)
5.5 Passing arrays to function.
5.6 Array applications - Finding maximum and minimum, Counting occurrences, Linear search, Sorting an array (Simple exchange sort, bubble sort), Merging two sorted arrays, Matrix operations (trace of matrix, addition, transpose, multiplication, symmetric, upper/ lower triangular matrix)

Reference Books:

1. How to Solve it by Computer, R.G. Dromey, Pearson Education.
2. Problem Solving and Programming Concept, Maureen Sprankle,7th Edition, Pearson Publication.

3. C: the Complete Reference, Schildt Herbert, 4th edition, McGraw Hill
4. A Structured Programming Approach Using C, Behrouz A. Forouzan, Richard F. Gilberg, Cengage Learning India
5. The 'C' programming language, Brian Kernighan, Dennis Ritchie, PHI
6. Programming in C ,A Practical Approach, Ajay Mittal , Pearson
7. Programming with C, B. Gottfried, 3rd edition, Schaum's outline Series, Tata McGraw Hill.
8. Programming in ANSI C, E. Balagurusamy, 7th Edition, McGraw Hill.

Semester- I
Paper - II

Course Type: Core Credit **Course Code: CS102**

Course Title : Database Management Systems

Teaching Scheme	No. of Credits	Examination Scheme
02 Hours / Week	2	IE : 15 Marks UE: 35 Marks

Prerequisites

- Basic Knowledge of file system, storing data in file system and Operations on sets

Course Objectives

- To understand the fundamental concepts of database.
- To understand user requirements and frame it in data model.
- To understand creations, manipulation and querying of data in databases.

Course Outcomes

On completion of the course, student will be able to–

- Solve real world problems using appropriate set, function, and relational models.
- Design E-R Model for given requirements and convert the same into database tables.
- Use SQL.

Course Contents

Chapter 1	**Introduction to DBMS**	**3 Hours**

1.1. Introduction
1.2. File system Vs DBMS
1.3. Levels of abstraction & data independence
1.4.Structure of DBMS (Roles of DBMS Users)
1.5. Users of DBMS Advantages of DBMS

Chapter 2	**Conceptual Design**	**11 Hours**

2.1. Overview of DB design process
2.2. Introduction to data models (E-R model, Relational model, Network model, Hierarchical model)
2.3. Conceptual design using ER data model (entities, attributes, entity sets, relations, relationship sets)
2.4. Constraints (Key constraints, Integrity constraints, referential integrity, unique constraint, Null/Not Null constraint, Domain, Check constraint, Mapping constraints)
2.5. Extended features – Specialization, Aggregation, Generalization
2.6. Pictorial representation of ER(symbols)
2.7. Structure of Relational Databases (concepts of a table)
2.8. DBMS Versus RDBMS
2.9. Case Studies on ER model

Chapter 3	**SQL**	**9 Hours**
3.1. Introduction to query languages 3.2. Basic structure 3.3. DDL Commands 3.4. DML Commands 3.5. Forms of a basic SQL query (Expression and strings in SQL) 3.6. Set operations 3.7. Aggregate Operators and functions 3.8. Date and String functions 3.9. Null values 3.10. Nested Subqueries 3.11 SQL mechanisms for joining relations (inner joins, outer joins and their types) 3.12 Views 3.13. Examples on SQL (case studies)		
Chapter 4	**Relational Database Design**	**7 Hours**
3.1. Introduction to Relational-Database Design (undesirable properties of a RDB design) 3.2. Functional Dependency(Basic concepts, F+, Closure of an Attribute set, Armstrong's axioms) 3.3. Concept of Decomposition 3.4. Desirable Properties of Decomposition (Lossless join, Lossy join, Dependency Preservation) 3.5. Concept of normalization, Normal Forms (1NF,2NF and 3NF), Examples 3.6 Keys Concept with Examples : Candidate Keys and Super Keys, Algorithm to find the super keys / primary key for a relation		
Reference Books:		
1. Database System Concepts, Henry F. Korth, Abraham Silberschatz, S.Sudarshan,ISBN:9780071289597,Tata McGraw-Hill Education 2. Database Management Systems ,RaghuRamakrishnan,ISBN:9780071254342,Mcgraw-hill higher Education 3. Database Management Systems, Raghu Ramakrishnan and Johannes Gehrke,McGraw-Hill Science/Engineering/Math; 3 edition, ISBN: 9780072465631 4. Database Systems, Shamkant B. Navathe, RamezElmasri,ISBN:9780132144988,PEARSON HIGHER EDUCATION 5. Beginning Databases with PostgreSQL: From Novice to Professional, Richard Stones, Neil Matthew, ISBN:9781590594780, Apress 6. PostgreSQL, Korry Douglas, ISBN:9780672327568, Sams 7. Practical PostgreSQL (B/CD),JohnWorsley, Joshua Drake,ISBN:9788173663925Shroff/O'reilly 8. Practical Postgresql , By Joshua D. Drake, John C Worsley **(**O'Reillypublications) 9**.** "An introduction to Database systems", Bipin C Desai, Galgotia Publications		

Semester- I
Paper - III

Course Type: Core Credit **Course Code:CS103**

Title : Practical course on Problem Solving using Computer and 'C' programming and Database Management Systems

Teaching Scheme	No. of Credits	Examination Scheme
3 Hrs / week	1.5	IE : 15 Marks UE: 35 Marks

Course Objectives

- To understand the program development life cycle.
- Solve simple computational problems using modular design and basic features of the 'C' language.
- Understand basic database management operations.
- Design E-R Model for given requirements and convert the same into database tables.

Course Outcomes:-

On completion of this course, students will be able to :

- Devise pseudocodes and flowchart for computational problems.
- Write, debug and execute simple programs in 'C'.
- Create database tables in postgreSQL.
- Write and execute simple, nested queries.

Guidelines :

Lab Book: The lab book is to be used as a hands-on resource, reference and record of assignment submission and completion by the student. The lab book contains the set of assignments which the student must complete as a part of this course.

Submission:

Problem Solving Assignments:

The problem solving assignments are to be submitted by the student in the form of a journal containing individual assignment sheets. Each assignment includes the Assignment Title, Problem statement, Date of submission, Assessment date, Assessment grade and instructors sign.

Programming Assignments:

Programs should be done individually by the student in the respective login. The codes should be uploaded on either the local server, Moodle, Github or any open source LMS. Print-outs of the programs and output may be taken but not mandatory for assessment.

DBMS Assignments:

For each problem/case study, the student must design the database model in the form of an E-R

diagram. Table design should be based on the same and must include proper constraints and integrity checks. The students have to create, populate the tables and then perform the activities specified in each of the assignments. A pool of databases will get created as student progresses through the assignments and these databases can be repeatedly used in subsequent assignments. A separate softcopy of the queries must be maintained for each assignment.

Assessment:

Continuous assessment of laboratory work is to be done based on overall performance and lab assignments performance of student. Each lab assignment assessment will be assigned grade/marks based on parameters with appropriate weightage. Suggested parameters for overall assessment as well as each lab assignment assessment include- timely completion, performance, innovation, efficient codes and good programming practices.

Operating Environment:

For 'C' Programming :
Operating system: Linux
Editor: Any linux based editor like vi, gedit etc.
Compiler : cc or gcc

For DBMS:
Operating System: Linux Operating system
DBMS: PostgreSQL
Language: SQL

Suggested List of Assignments:

A) Problem Solving and C programming:

Assignment 1.
Problem Solving using Pseudo code and Flowchart, Simple programs, Understanding errors and error handling.

Assignment 2.
Decision Making Control Structures.

Assignment 3.
Loop Control Structures

Assignment 4.
Functions (User Defined functions, Library functions and Recursion).

Assignment 5.
Arrays (1-D and 2-D).

B) Database Management Systems

Assignment 1.

To create simple tables with only the primary key constraint (as a table level constraint & as a field level constraint) (include all data types)

Assignment 2.

To create more than one table, with referential integrity constraint, PK constraint.

Assignment 3.

To create one or more tables with following constraints, in addition to the first two constraints (PK & FK)

a. Check constraint
b. Unique constraint
c. Not null constraint

Assignment 4.

To drop a table, alter schema of a table, insert / update / delete records using tables created in previous Assignments. (use simple forms of insert / update / delete statements)

Assignment 5.

To query the tables using simple form of select statement Select <field-list> from table [where <condition> order by <field list>] Select <field-list, aggregate functions > from table [where <condition> group by <> having <> order by <>]

Assignment 6.

To query table, using set operations (union, intersect)

Assignment 7.

To query tables using nested queries (use of 'Except', exists, not exists, all clauses

Assignment 8.

To create views.

Books: Laboratory handbook prepared by the University.

Semester- II
Paper - I

Course Type: Core Credit **Course Code: CS201**

Course Title : Advanced 'C' Programming

Teaching Scheme	No. of Credits	Examination Scheme
2 Hours / Week	2	IE : 15 Marks UE: 35 Marks

Prerequisites

- Problem Solving tools like algorithms, flowcharts and pseudocodes.
- Basic knowledge of 'C' language.

Course Objectives :-

- To study advanced concepts of programming using the 'C' language.
- To understand code organization with complex data types and structures.
- To work with files.

Course Outcomes:- Student will be able to :-

- Develop modular programs using control structures, pointers, arrays, strings and structures
- Design and develop solutions to real world problems using C.

Course Contents

Chapter 1	**Pointers**	**8 Hours**

1.1. Introduction to Pointers.
1.2. Declaration, definition, initialization, dereferencing.
1.3. Pointer arithmetic.
1.4. Relationship between Arrays & Pointers- Pointer to array, Array of pointers.
1.5. Multiple indirection (pointer to pointer).
1.6. Functions and pointers- Passing pointer to function, Returning pointer from function, Function pointer.
1.7. Dynamic memory management- Allocation(malloc(),calloc()), Resizing(realloc()), Releasing(free()).,
1.8. Memory leak, dangling pointers.
1.9. Types of pointers.

Chapter 2	**Strings**	**6 Hours**

2.1 String Literals, string variables, declaration, definition, initialization.
2.2 Syntax and use of predefined string functions
2.3 Array of strings.
2.4. Strings and Pointers
2.5. Command line arguments.

Chapter 3	Structures And Unions.	8 Hours

3.1. Concept of structure, definition and initialization, use of typedef.
3.2. Accessing structure members.
3.3. Nested Structures
3.4. Arrays of Structures
3.5. Structures and functions- Passing each member of structure as a separate argument, Passing structure by value / address.
3.6. Pointers and structures.
3.7. Concept of Union, declaration, definition, accessing union members.
3.8. Difference between structures and union.

Chapter 4	File Handling	6 Hours

4.1. Introduction to streams.
4.2. Types of files.
4.3. Operations on text files.
4.4. Standard library input/output functions.
4.5. Random access to files.

Chapter 5	Preprocessor	2 Hours

6.1. Role of Preprocessor
6.2. Format of preprocessor directive
6.3. File inclusion directives (#include)
6.4. Macro substitution directive, argumented and nested macro
6.5. Macros versus functions

Reference Books:

1. C: the Complete Reference, Schildt Herbert, 4th edition, McGraw Hill
2. A Structured Programming Approach Using C, Behrouz A. Forouzan, Richard F. Gilberg, Cengage Learning India
3. The 'C' programming language, Brian Kernighan, Dennis Ritchie, PHI
4. Programming in C ,A Practical Approach, Ajay Mittal , Pearson
5. Programming with C, B. Gottfried, 3rd edition, Schaum's outline Series, Tata McGraw Hill.
6. Programming in ANSI C, E. Balagurusamy, 7th Edition, McGraw Hill.

Semester- II
Paper - II

Course Type: Core Credit **Course Code: CS202**

Course Title : Relational Database Management Systems

Teaching Scheme	No. of Credits	Examination Scheme
2 Hours / Week	2	IE : 15 Marks UE: 35 Marks

Prerequisites

- Basic Knowledge of DBMS
- Knowledge of SQL Queries
- Basics of relational design
- Basics of ER model

Course Objectives

- To teach fundamental concepts of RDBMS (PL/PgSQL)
- To teach database management operations
- Be familiar with the basic issues of transaction processing and concurrency control
- To teach data security and its importance

Course Outcomes

On completion of the course, student will be able to–

- Design E-R Model for given requirements and convert the same into database tables.
- Use database techniques such as SQL & PL/SQL.
- Explain transaction Management in relational database System.
- Use advanced database Programming concepts

Course Contents

Chapter 1	**Relational Database Design Using PLSQL**	**8 Hours**

1.1 Introduction to PLSQL
1.2 PL/PgSqL: Datatypes, Language structure
1.3 Controlling the program flow, conditional statements, loops
1.4 Stored Procedures
1.5 Stored Functions
1.6 Handling Errors and Exceptions
1.7 Cursors
1.8 Triggers

Chapter 2	Transaction Concepts and concurrency control	10 hours

2.1 Describe a transaction, properties of transaction, state of the transaction.
2.2 Executing transactions concurrently associated problem in concurrent execution.
2.3 Schedules, types of schedules, concept of Serializability, Precedence graph for Serializability.
2.4 Ensuring Serializability by locks, different lock modes, 2PL and its variations.
2.5 Basic timestamp method for concurrency, Thomas Write Rule.
2.6 Locks with multiple granularity, dynamic database concurrency (Phantom Problem).
2.7 Timestamps versus locking.
2.8 Deadlock and deadlock handling - Deadlock Avoidance(wait-die, wound-wait), Deadlock Detection and Recovery (Wait for graph).

Chapter 3	Database Integrity and Security Concepts	6 Hours

3.1 Domain constraints
3.2 Referential Integrity
3.3 Introduction to database security concepts
3.4 Methods for database security
  3.4.1Discretionary access control method
  3.4.2Mandatory access control
  3.4.3. Role base access control for multilevel security.
3.5 Use of views in security enforcement.
3.6 Overview of encryption technique for security.
3.7 Statistical database security.

Chapter 4	Crash Recovery	4 Hours

4.1 Failure classification
4.2 Recovery concepts
4.3 Log base recovery techniques (Deferred and Immediate update)
4.4 Checkpoints, Relationship between database manager and buffer cache. Aries recovery algorithm.
4.5 Recovery with concurrent transactions (Rollback, checkpoints, commit)
4.6 Database backup and recovery from catastrophic failure

Chapter 5	Other Databases	2 Hours

5.1 Introduction to Parallel and distributed Databases
5.2 Introduction to Object Based Databases
5.3 XML Databases
5.4 NoSQL Database
5.5 Multimedia Databases
5.6 Big Data Databases

Reference Books:

1. Database System Concepts, By Silberschatz A., Korth H., Sudarshan S., 6th Edition, McGraw Hill Education
2. Database Management Systems, Raghu Ramakrishnan, Mcgraw-Hill Education
3. Database Systems, Shamkant B. Navathe, Ramez Elmasri, PEARSON HIGHER EDUCATION
4. Fundamentals of Database Systems, By: Elmasri and Navathe, 4th Edition Practical PostgreSQL O'REILLY
5. Database Management Systems,RaghuRamakrishnan and JohannesGehrke,McGraw-Hill Science/Engineering/Math; 3 edition, ISBN: 9780072465631
6. NoSQL Distilled, Pramod J. Sadalage and Martin Fowler, Addison Wesley
7. An Introduction to Database Systems", C J Date, Addison-Wesley
8. Database Systems : Concepts, Design and Application", S.K.Singh, Pearson, Education
9. NoSQL Distilled A Brief Guide to the Emerging World of Polyglot Persistence : by Pramod J. Sadalage, Martin Fowler, Addison-Wesley, Pearson Education, Inc.
10. MongoDB: The Definitive Guide , Kristina Chodorow, Michael Dirolf, O'Reilly Publications

Semester- II
Paper - III

Course Type: Core Credit **Course Code:CS203**

Title : Practical Course on Advanced 'C' Programming and Relational Dstabase Management Systems

Teaching Scheme	No. of Credits	Examination Scheme
3 Hours / week	1.5	IE : 15 Marks UE: 35 Marks

Course Objectives

- To solve real world computational problems.
- To perform operations on relational database management systems.

Course Outcomes:-

On completion of this course, students will be able to :

- Write, debug and execute programs using advanced features in 'C'.
- To use SQL & PL/SQL.
- To perform advanced database operations.

Guidelines :

Lab Book: The lab book is to be used as a hands-on resource, reference and record of assignment submission and completion by the student. The lab book contains the set of assignments which the student must complete as a part of this course.

Submission:

Programming Assignments:

Programs should be done individually by the student in respective login. The codes should be uploaded on either the local server, Moodle, Github or any open source LMS. Print-outs of the programs and output may be taken but not mandatory for assessment.

RDBMS Assignments:

For each problem/case study, the student must design the database model in the form of an E-R diagram. Table design should be based on the same and must include proper constraints and integrity checks. The students have to create, populate the tables and then perform the activities specified in each of the assignments. A separate softcopy of the table creation statements and queries must be maintained for each assignment.

Assessment

Continuous assessment of laboratory work is to be done based on overall performance and lab assignments performance of student. Each lab assignment assessment will be assigned grade/marks based on parameters with appropriate weightage. Suggested parameters for overall

assessment as well as each lab assignment assessment include- timely completion, performance, innovation, efficient codes and good programming practices.

Operating Environment:

For 'C' Programming :
Operating system: Linux
Editor: Any linux based editor like vi, gedit etc.
Compiler : cc or gcc

For DBMS:
Operating System: Linux Operating system
DBMS: PostgreSQL 11 and higher
Language: PL/SQL

Suggested List of Assignments:

A) Advanced C Programming:

Assignment 1.

Simple Pointers.

a) Pointer initialization and use of pointers.
b) Pointer Arithmetic.

Assignment 2.

Dynamic Memory Allocation.

Assignment 3.

String handling using standard library functions.

Assignment 4.

Structure and Unions.

Assignment 5.

File Handling.

Assignment 6.

C Preprocessors.

B) Relational Database Management Systems:

Assignment 1: Stored Procedure

1) A Simple Stored Procedure
2) A Stored Procedure with IN, OUT and IN/OUT parameter

Assignment 2: Stored Function

1) A Simple Stored Function
2) A Stored Function that returns
3) A Stored Function recursive

Assignment 3 : Cursors

1) A Simple Cursor
2) A Parameterize Cursor

Assignment 4 : Exception Handling

1) Simple Exception- Raise Debug Level Messages
2) Simple Exception- Raise Notice Level Messages
3) Simple Exception- Raise Exception Level Messages

Assignment 5 : Triggers

1) Before Triggers (insert, update, delete)
2) After Triggers (insert, update, delete)

Books: Laboratory handbook prepared by the University.