VEER NARMAD SOUTH GUJARAT UNIVERSITY – SURAT

Bachelor of Computer Application

Program Structure	Semester-wise break up for the courses is given below:

SEMESTER - 3

Course Code	Title	Teaching per week		Course Credits	University Examination		Internal Marks	Total Marks
		Theory	Practical		Duration	Marks		
301	Statistical Methods	2	0	2	3 Hrs	70	30	100
302	Software Engineering-I	3	0	3	3 Hrs	70	30	100
303	Relational Database Management System (RDBMS)	4	0	4	3 Hrs	70	30	100
304	Data Structures	4	0	4	3 Hrs	70	30	100
305	Object Oriented Programming	4	0	4	3 Hrs	70	30	100
306	Practical	0	12	6	5 Hrs	140	60	200
	Foundation Elective (to be selected from NCC / NSS / Saptadhara)	0	2	2				
Total		17	14	25		490	210	700

For Practical:

1. Batch Size – 30 Maximum
2. In case of more than 10 students in a batch, separate batch should be considered.
3. The journal should be certified by the concerned faculty and by the Head of the Department, failing which the student should not be allowed to appear for External Practical Examination.

SEMESTER - 4

Course Code	Title	Teaching per week		Course Credits	University Examination		Internal Marks	Total Marks
		Theory	Practical		Duration	Marks		
401	Information System	2	0	2	3 Hrs	70	30	100
402	Software Engineering-II	3	0	3	3 Hrs	70	30	100
403	Java Programming	4	0	4	3 Hrs	70	30	100
404	.NET Programming	4	0	4	3 Hrs	70	30	100
405	Web Designing	4	0	4	3 Hrs	70	30	100
406	Practical	0	12	6	5 Hrs	140	60	200
	Foundation Elective (to be selected from NCC / NSS / Saptadhara)	0	2	2				
Total		17	14	25		490	210	700

For Practical:

1. Batch Size – 30 Maximum
2. In case of more than 10 students in a batch, separate batch should be considered.
3. The journal should be certified by the concerned faculty and by the Head of the Department, failing which the student should not be allowed to appear for External Practical Examination.

Program Passing Rules	As per University rules.

Course: 301: Statistical Methods

Course Code	301
Course Title	Statistical Methods
Credit	2
Teaching per Week	2 Hrs
Minimum weeks per Semester	15 (Including Class work, examination, preparation etc.)
Review / Revision	June 2018
Purpose of Course	To develop statistical problems solving abilities relevant to Computer Science.
Course Objective	1. To make students understand various statistical methods. 2. To develop the ability to compute descriptive statistics including diagrammatic representation and interpretation. 3. To be able to carry out simple linear regression analysis.
Pre-requisite	None
Course Out come	Ability to use computers to analyse data.
Course Content	**Unit 1. Introduction and Presentation of statistical data** 1.1. Types of variables 1.2. Univariate, bivariate and multivariate data 1.3. Univariate and bivariate frequency distributions **Unit 2. Measure of central tendency-mean, median and mode** **Unit 3. Measures of dispersion (absolute as well as relative)** 3.1. Mean deviation 3.2. Standard deviation 3.3. Coefficient of mean deviation and coefficient of variation **Unit 4. Correlation** 4.1. Introduction 4.2. Types of correlation and scatter diagrams 4.3. Rank correlation coefficient **Unit 5. Regression** 5.1. Concept of dependent and independent variables 5.2. Introduction to liner regression 5.3. Line of regression (with one independent variable) Methods should be explained conceptually and corresponding examples should be given. No proof should be given to any of the methods.
Reference Book	1. Introduction to mathematical statistics, Hogg R V & Craig A L - Tata McGraw Hill 2. An introduction to the theory of statistics, Yule U G & Kendall MG – C. Griffin & Co. 3. Statistical Methods, S. P. Gupta – Sultan Chand & Co
Teaching Methodology	Class Work, Discussion, Self-Study, Seminars and/or Assignments
Evaluation Method	30% Internal assessment. 70% External assessment.

Course: 302: Software Engineering-I

Course Code	302
Course Title	Software Engineering - I
Credit	3
Teaching per Week	3 Hrs
Minimum weeks per Semester	15 (Including Class work, examination, preparation etc.)
Review / Revision	June 2018
Purpose of Course	Computer software engineers apply the principles and techniques of computer science, engineering, and mathematical analysis to the design, development, testing, and evaluation of the software and the systems that enable computers to perform their many applications.
Course Objective	1. To make students understand how to engineer the software. 2. To make students understand various components of software process model and their working. 3. To make students understand the importance of requirement analysis. 4. To make students understand various approaches of system design.
Pre-requisite	Prior knowledge of some software.
Course Out come	After studying this, students will be able to understand how software is engineered and importance of various aspects of software engineering. This course will also help students appreciate the role of various design principles. After successful completion students will be able to perform requirement analysis and system design for their applications.
Course Content	**Unit 1.** Introduction 1.1 What is software? 1.2 Software characteristics. 1.3 Software Engineering: definition. **Unit 2.** Software Engineering 2.1 Software Applications, Myths. 2.2 Software Engineering: Generic View. **Unit 3.** Software Process models 3.1 Introduction of Waterfall model. 3.2 Prototype model. 3.3 Spiral Model 3.4 Incremental Model **Unit 4.** Requirement analysis 4.1 Introduction. 4.2 Current Application Analysis. 4.3 Requirement gathering techniques & Fact Finding, Recording Outcome. 4.4 DFD, Data Dictionary and Process Specification. 4.5 Importance of Requirement Specifications. 4.6 Software Requirement Specification Document.

	Unit 5. System Design 5.1 Design model. 5.2 Principal and Concepts. 5.3 Functional Independence. 5.4 Module & Sequence. 5.5 Effectiveness of Modular Design. 5.6 Mapping of Requirements into Design. 5.7 Design Documentation.
Reference Books	1. Software Engineering - A Practitioners' approach, R. S. Pressman – McGraw Hill. 2. Software Engineering concepts, Richard Fairley – McGraw Hill. 3. An Integrated Approach to Software Engineering, Pankaj Jalota – Narosa. 4. Software Engineering a Concise Study, Kelkar – PHI. 5. Fundamentals of Software Engineering, 4th Edition, Rajib Mall – PHI. 6. Software Engineering, Ian Sommerville - Pearson Education. 7. System Analysis & Design in changing world, Satzinger, Jackson, Burd – Course Technology. 8. System Analysis, Design & Introduction to S/W Engineering, Prof. S. Parthasathy & Prof. B. W. Khalkar – Master Academy, Nashik. 9. System Analysis & Design, Elias M – Galgotia Publications. 10. System Analysis & Design, Richard Fairley - Galgotia Publications.
Teaching Methodology	Class Work, Discussion, Self-Study, Seminars and/or Assignments
Evaluation Method	30% Internal assessment. 70% External assessment.

Course: 303: Relational Database Management System

Course Code	303
Course Title	Relational Database Management System
Credit	4
Teaching per Week	4 Hrs
Minimum weeks per Semester	15 (Including Class work, examination, preparation etc.)
Review / Revision	June 2018
Purpose of Course	Give fundamental knowledge of Relational Database. The course also includes SQL & PL/SQL.
Course Objective	1. To make students understand Oracle architecture. 2. To make students understand various components of database like Index Triggers etc. 3. To make students understand the importance of database in real world applications. 4. To make students aware of extracting the data in different ways.
Pre-requisite	Basic knowledge of Database Management System (DBMS).
Course Out come	After learning this subject, students will know how to store, retrieve and administer the data easily & efficiently.
Course Content	**Unit 1. Codd's Rules** **Unit 2. SQL** 2.1. Oracle Data Types 2.2. Oracle DDL Commands (Create Table, Alter Table, Drop Table), DML Commands (Insert, Update, Delete, Select) and TCL Commands (Commit, Rollback, SavePoint) Statements with integrity constraints 2.3. Special Operators (IN, NOT IN, EXISTS, LIKE) 2.4. Oracle Functions 2.4.1. Scalar Functions (String Functions, Numeric Functions, Date Functions, Conversion Functions) 2.4.2. Aggregate Functions 2.5. Range Searching and Pattern Matching 2.6. Manipulating Dates 2.7. Joins 2.7.1. Inner Join 2.7.2. Outer Join (Left, Right, Full) 2.7.3. Cross Join 2.8. Sub Queries 2.9. Using Union, Intersection and Minus Clauses 2.10. Indexes (Create index, Drop Index, Types of Index) 2.11. Views (Read-only view, Updatable view) 2.12. Sequences **Unit 3. PL/SQL** 3.1. PL/SQL Block Structure 3.1.1. Using Variables, Constants and Data Types 3.1.2. User Defined Record 3.1.3. Assigning Values to Variables 3.1.4. Control Statements (IF…THEN statement, Loop,

	FOR...Loop, While Loop) 3.2. Cursor (Explicit, Implicit) 3.3. Error handling in PL/SQL 3.3.1. Inbuilt Exceptions 3.3.2. User Defined Exception 3.4. Stored and Local Procedures & Functions **Unit 4. Database Triggers** 4.1. Definition of Trigger 4.2. Statement level Triggers 4.3. Row level Triggers **Unit 5. Database Packages** 5.1. Introduction 5.2. Components of Package 5.3. Create and Invoke Package
Reference Book	1. The Complete Reference, George Koch, Kevin Loney – Oracle Press 2. Database Management System, Oracle, SQL and PL/SQL, 2nd ed., Das Gupta & Radha Krishna, PHI 3. Oracle 9 PL/SQL Programming, Scott Urman – Oracle Press 4. Oracle SQL: The Essential Reference, David C. Kreines – O'Reilly 5. SQL, PL/SQL: The Programming Language of Oracle, Ivan Bayross – BPB 6. Oracle PL/SQL Programming – Feuerstein & Peribyl – SPD O'Reilly 7. Learning Oracle SQL and PL/SQL: A Simplified Guide, Chatterjee – PHI
Teaching Methodology	Class Work, Discussion, Self-Study, Seminars and/or Assignments
Evaluation Method	30% Internal assessment. 70% External assessment.

Course: 304: Data Structures

Course Code	304
Course Title	Data Structures
Credit	4
Teaching per Week	4 Hrs
Minimum weeks per Semester	15 (Including class work, examination, preparation etc.)
Review / Revision	June 2018
Purpose of Course	It is important for a computer programmer to understand the storage representation and implementation of various data structures used in a computer program. This helps a programmer to use various data structures efficiently which in turn makes the program efficient. This course introduces various data structures, their storage representation & implementation.
Course Objective	1. Get detailed knowledge of basic data structures, role and importance of data structures in computer programming. 2. Distinguish the key difference between storage & implementation of various data structures. 3. Recognize the problem properties and determine the use of appropriate data structures in different real-world applications. 4. Learn and compare various searching & sorting techniques.
Pre-requisite	This course requires 1. Problem-solving, design and implementation skills. 2. Basic knowledge of programming language.
Course outcome	Students will be able to 1. Implement various operations of data structures and utilities using algorithm. 2. Select appropriate methods for organizing data files and implement file-based data structures.
Course Content	**Unit 1.** Introduction to data structures 1.1 Definition 1.2 Types of Data Structure 1.2.1 Primitive Data Structures 1.2.2 Non-primitive Data Structure (Linear and Non-Linear) 1.3 Storage representation of primitive data structure (integer and character) **Unit 2.** Non-Primitive Linear Data Structures 2.1 Arrays – its storage structure and Operations (insertion and deletion) 2.2 Stack 2.2.1 Stack operations 2.2.2 Applications of Stack (Recursion and Polish notations) 2.3 Queue 2.3.1 Types of Queues (Simple, Circular, Double-ended and Priority) 2.3.2 Operations on Queue 2.3.3 Application of Queue (Simulation) 2.4 Linked list 2.4.1 Types of Linked lists (Singly, Doubly, Circular)

	2.4.2 Operations on Linked list 2.4.3 Applications of Linked list (Polynomial manipulation) **Unit 3.** Non-Primitive Non-Linear Data Structures 3.1 Definition of Graph 3.2 Concept and Definition of Tree 3.3 Types of Binary Tree (Ordinary/Simple, Strictly and Complete Binary tree) 3.4 Operations on Binary tree (Traversals, Insertion and Deletion) 3.5 Storage representation of Binary tree (Linked, Sequential and Threaded) 3.6 Binary search tree 3.7 Application of tree (Manipulation of arithmetic expression) **Unit 4.** Searching & Sorting Techniques 4.1 Introduction 4.2 Searching Techniques (Sequential and Binary) 4.3 Types of Sorting Techniques (Insertion, Selection, Quick, 2-Way Merge and Bubble) **Unit 5.** Balance trees 5.1 Introduction 5.2 Balance trees 5.2.1 AVL tree 5.2.2 2-3 tree
Reference Books	1. An Introduction to Data Structures with applications, Trembley – Tata McGraw Hill. 2. Algorithms – Data structure programs, Wirth Niclaus - PHI. 3. Data structures – A Programming Approach with C, Dharmender Singh Kushwaha and Arun Kumar Misra – PHI. 4. Fundamentals of Data structures, Horwitz E. and Sahni – Computer Science Press 5. Schaum's outline of Data Structure with C++, John R. H. - Tata McGraw Hill. 6. Expert Data Structure with C, R. B. Patel - Khanna Publication 7. Data structures - a Pseudocode approach with C++, Richard F. Gilberg and Behrouz A. Forouzan - Thomson books
Teaching Methodology	Class Work, Discussion, Self-Study, Seminars and/or Assignments
Evaluation Method	30% Internal assessment. 70% External assessment.

Course: 305: Object Oriented Programming

Course Code	305
Course Title	Object Oriented Programming
Credit	4
Teaching per Week	4 Hrs
Minimum weeks per Semester	15 (Including Class work, examination, preparation etc.)
Review / Revision	June 2018
Purpose of Course	Understand Object Oriented Programming Concepts and skills necessary for developing programs using C++.
Course Objective	1. C++ runs on a variety of platforms, such as Windows, Mac OS, and the various versions of UNIX. This course has been designed for the beginners to help them understand basic to advanced concepts related to C++ Programming language. 2. To make students understand the importance of OOP methodology. 3. To make students understand exception handling and file handling. 4. To make students understand various types of OOP techniques.
Pre-requisite	Basic understand of Computer program and C programming language.
Course Out come	After studying this, students will be able to understand how OOP principles work and importance of various coding techniques of OOP. This course will also help students appreciate the role of Exception handling and File handling techniques. After successful completion students will be able to follow programming methodology and will understand how to apply it for their application.
Course Content	**Unit 1. Principles of Object Oriented Programming** 1.1. Procedure Oriented Programming Vs Object Oriented Programming 1.2. Basic concepts of Object Oriented Programming (Encapsulation, Polymorphism etc) 1.3. Benefits of Object Oriented Programming 1.4. Structure & Classes 1.5. Encapsulation and Data Hiding 1.6. Constructors 1.7. Friend Function 1.8. Inline Function 1.9. Dynamic Object Creation & Destruction 1.10. Destructor **Unit 2. Object Oriented Properties** 2.1. Introduction to Object Oriented Properties 2.2. Abstraction 2.3. Inheritance 2.3.1. Type of Inheritance 2.3.2. Constructors and Destructor Calls during Inheritance 2.3.3. Abstract Class **Unit 3. Polymorphism** 3.1 Static Polymorphism 3.1.1 Operator Overloading

	3.1.2 Function Overloading and Type Conversion 3.2 Dynamic Polymorphism 3.2.1 Overriding 3.2.2 Virtual Function **Unit 4. Data Files** 4.1 Manipulators (In-Built, User Defined) 4.2 File Modes 4.3 File Functions 4.4 Error Handling During File Operation **Unit 5. Exception Handling** 5.1 Introduction to Exception 5.2 Try ... Catch
Reference Book	1. Let us C++, Yaswant Kanitkar - TMH Publication 2. Programming with C++, E Balaguruswamy - BPB Publication 3. C++ and Object-Oriented Programming Paradigm, Jana - PHI 4. The Complete Reference C++, Herbert Schildt - TMH 5. The C++ Programming Language, Stroustrup – Addison Wesley 6. OOP in Turbo C++, Robert Lafore - Galgotia Publication 7. C++ Primer, Lippman – Addison Wesley 8. Object Oriented Programming Fundamentals & Applications, Probal Sengupta – PHI
Teaching Methodology	Class Work, Discussion, Self-Study, Seminars and/or Assignments
Evaluation Method	30% Internal assessment. 70% External assessment.

Course: 306: Practical

Course Code	306
Course Title	Practical
Credit	6
Teaching per Week	12 Hrs
Minimum weeks per Semester	15 (Including Class work, examination, preparation etc.)
Review / Revision	June 2018
Purpose of Course	Through practical implementation the students can understand & learn computer programming in a better way.
Course Objective	The Objective of this course is to make students practically learn the concepts taught in Paper nos. 303, 304, 305.
Pre-requisite	Programming in C and DBMS.
Course Out come	After completion of this course, the students will be able to implement practical problems related to Data Structures, RDBMS and Object-Oriented programming.
Course Content	Practical based on Papers 303, 304 and 305
Reference Book	As per papers 303, 304 and 305.
Teaching Methodology	Lab. Work
Evaluation Method	30% Internal assessment. 70% External assessment.