SWAMI RAMANAND TEERTH MARATHWADA UNIVERSITY, NANDED

(NAAC Re-accredited with 'A' Grade)

School of Computational Sciences

स्वामी रामानंद तीर्थ मराठवाडा विद्यापीठ, नांदेड.

CURRICULUM FRAMEWORK AND SYLLABUS

FOR OUTCOME BASED EDUCATION IN

M.Sc. (C.S.) Degree Program

FOR THE STUDENTS ADMITTED FROM THE

ACADEMIC YEAR 2019-2020 ONWARDS

www.srtmun.ac.in

BOS meeting approved: __/__/2019

Approved in ____ Academic Council

Program Educational Objectives (PEO)

Post graduates of M.Sc.(CS) program will be

PEO1: Utilizing strong technical aptitude and domain knowledge to develop smart software solutions for the upliftment of society.

PEO2: Applying research and entrepreneurial skills augmented with a rich set of communication, teamwork and leadership skills to excel in their profession.

PEO3: Showing continuous improvement in their professional career through life-long learning, appreciating human values and ethics.

Graduate Attributes for M.Sc.(CS) Program (GA)

1. Computational Knowledge:

Apply domain knowledge appropriate for the computing specialization to the abstraction and conceptualization of computing models from defined problems and requirements.

2. Problem Analysis:

Identify, formulate, research literature, and solve *complex* computing problems reaching substantiated conclusions using fundamental principles of computing sciences.

3. Design /Development of Solutions:

Design and evaluate solutions for *complex* computing problems that meet specified needs with appropriate consideration for cultural, societal, and environmental considerations.

4. Conduct Investigations of Complex Computing Problems:

Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. Modern Tool Usage:

Create, select, adapt and apply appropriate techniques, resources, and modern computing tools to *complex* computing activities, with an understanding of the limitations.

6. Professional Ethics:

Understand and commit to professional ethics and cyber regulations, responsibilities, and norms of professional computing practice.

7. Life-long Learning:

Recognize the need, and have the ability, to engage in independent learning for continual development as a computing professional.

8. Project management and finance:

Demonstrate knowledge and understanding of the computing and management principles and apply these to one’s own work, to manage projects and in multidisciplinary environments.

9. Communication Efficacy:

Communicate effectively with the computing community, and with society at large, about *complex* computing activities by being able to comprehend and write effective reports, design documentation, make effective presentations, and give and understand clear instructions.

10. Societal and Environmental Concern:

Understand and assess societal, environmental, health, safety, legal, and cultural issues within local and global contexts, and the consequential responsibilities relevant to professional computing practice.

11. Individual and Team Work:

Function effectively as an individual and as a member or leader in diverse teams and in multidisciplinary environments.

12. Innovation and Entrepreneurship

Identify a timely opportunity and using innovation to pursue that opportunity to create value and wealth for the betterment of the individual and society at large.

Program Outcomes (PO) for M.Sc.(CS) (2019-2020)

On completion of M.Sc.(CS) program, the students are expected to

PO1: Apply knowledge of computing fundamentals, computing specialization, mathematics, and domain knowledge appropriate for the computing specialization to the abstraction and conceptualization of computing models from defined problems and requirements.

PO2: Identify, formulate, research literature, and solve complex computing problem searching substantiated conclusions using fundamental principles of mathematics, computing sciences, and relevant domain disciplines.

PO3: Design and evaluate solutions for complex computing problems, and design and evaluate systems, components, or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal, and environmental considerations.

PO4: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: Create, select, adapt and apply appropriate techniques, resources, and modern computing tools to complex computing activities, with an understanding of the limitations.

PO6: Understand and commit to professional ethics and cyber regulations, responsibilities, and norms of professional computing practice.

PO7: Recognize the need, and have the ability, to engage in independent learning for continual development as a computing professional.

PO8: Demonstrate knowledge and understanding of the computing and management principles and apply these to one**'**s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO9: Communicate effectively with the computing community, and with society at large, about complex computing activities by being able to comprehend and write effective reports, design documentation, make effective presentations, and give and understand clear instructions.

PO10: Understand and assess societal, environmental, health, safety, legal, and cultural issues within local and global contexts, and the consequential responsibilities relevant to professional computing practice.

PO11: Function effectively as an individual and as a member or leader in diverse teams and in multidisciplinary environments.

PO12: Identify a timely opportunity and using innovation to pursue that opportunity to create value and wealth for the betterment of the individual and society at large.

CBCS Revised Syllabus w.e.f AY: 2019-2020

Program: M.Sc.(Computer Science) (Campus) School of Computational Sciences

Sr. No	Course category	Course Code	Course Title	Internal credits	External credits	Total credits
First Semester						
1.	**Core Subjects**	NCS-101	Computer System Organization	2	2	4
2		NCS-102	Introduction to Databases	2	2	4
3		NCS-103	Mathematical Foundations for Computer Science	2	2	4
Choose any one from below elective subjects						
4	**Elective Subject**	NCS-104 A	Programming Language Concepts	2	2	4
		NCS-104 B	Object Oriented Programming			
Practical /Lab						
5	**Lab / Practical**	NCS-105	Lab-1:DBMS	1	1	2
		NCS-106	Lab-2: OOP	1	1	2
6	**Open Elective**	NCS-107A	University recognized MOOC (NPTEL / SWAYAM / others) OR Intra / Inter Departmental OR Intra / Inter School OR	4	0	4
		NCS-107 B	Information Communication Technology (ICT)			
7	**Skill based Activity**	NCS-108	SK-01	1	0	1
	Total credits					**25**

***NCS- Nanded Campus Computer Science**

Sr. No	Course category	Course Code	Course Title	Internal credits	External credits	Total credits
Second Semester						
1.	**Core Subjects**	NCS-201	Operating System Concepts	2	2	4
2		NCS-202	Elementary Data Structures and Algorithms	2	2	4
3		NCS-203	Programming in Java	2	2	4
Choose any one from below elective subjects						
4	**Elective Subject**	NCS-204 A	System Analysis and Design	2	2	4
		NCS-204 B	Data Communications			
Practical /Lab						
5	**Lab / Practical**	NCS-205	Lab-3: Data Structures	1	1	2
		NCS-206	Lab-4: Java Programming	1	1	2
6	**Open Elective**	NCS-207A	University recognized MOOC (NPTEL / SWAYAM / others) OR Intra / Inter Departmental OR Intra / Inter School OR	4	0	4
		NCS-207 B	Introduction to Linux			
7	**Skill based Activity**	NCS-208	SK-02	1	0	1
	Total credits					**25**

CBCS Revised Syllabus w.e.f AY: 2019-2020

Program: M.Sc.(Computer Science) (Campus) School of Computational Sciences

Sr. No	Course category	Course Code	Course Title	Internal credits	External credits	Total credits
Third Semester						
1.	**Core Subjects**	NCS-301	Windows Programming	2	2	4
2		NCS-302	Computer Networks	2	2	4
3		NCS-303	Compiler Designing	2	2	4
Choose any one from below elective subjects						
4	**Elective Subject**	NCS-304 A	Data Sciences	2	2	4
		NCS-304 B	Digital Image Processing			
Practical /Lab						
5	**Lab / Practical**	NCS-305	Lab-5: Windows Programming	1	1	2
		NCS-306	Lab-6: Based on Elective Subjects	1	1	2
6	**Open Elective**	NCS-307A	University recognized MOOC (NPTEL / SWAYAM / others) OR Intra / Inter Departmental OR Intra / Inter School OR	4	0	4
		NCS-307 B	Introduction to Web Technologies			
7	**Skill based Activity**	NCS-308	SK-03: Seminar Presentation Activity	1	0	1
	Total credits					**25**

Sr. No	Course category	Course Code	Course Title	Internal credits	External credits	Total credits
Fourth Semester						
1.	**Core Subjects**	NCS-401	Mobile Application Development	2	2	4
2		NCS-402	Advanced Databases and Administration	2	2	4
3		NCS-403	Major Project development Activity	0	4	4
Choose any one from below elective subjects						
4	**Elective Subject**	NCS-404 A	Internet of Things (IoT)	2	2	4
		NCS-404 B	Big Data Analytics			
Practical /Lab						
5	**Lab / Practical**	NCS-405	Lab-7: Mobile Application Development	1	1	2
		NCS-406	Lab-8: Advanced Databases	1	1	2
6	**Open Elective**	NCS-407A	University recognized MOOC (NPTEL / SWAYAM / others) OR Intra / Inter Departmental OR Intra / Inter School OR	4	0	4
		NCS-407 B	Programming in Python			
7	**Skill based Activity**	NCS-408	SK-04	1	0	1
	Total credits					**25**

CBCS Revised Syllabus w.e.f AY: 2019-2020

Program: M.Sc.(Computer Science) (Campus) School of Computational Sciences

Course Code:	**NCS-101**	**Course Name:** Computer System Organization	**Credits: 4**
Course Objectives:			
Student need to be understood by looking inside how computer architecture is build. Study of various components as building block Architecture of different configuration for different requirement or problem size Memory and IO related interfacing			
Course Outcome:			
Solve problems based on computer arithmetic Explain processor structure and its function Understating micro programming Understand concepts related to memory and IO mapping Design and analysis of memory and IO system			
Unit-1:	Basic Structure of Computers		
Functional units, basic operational concepts, Bus structures, Software performance, Memory locations and addresses, Memory operations, Instruction and instruction sequencing Addressing modes, Assembly language, Basic I/O operations, Stacks and queues.			
Unit-2:	Arithmetic Unit		
Addition and subtraction of signed numbers, Design of fast adders, Multiplication of positive numbers, Signed operand multiplication and fast multiplication, Integer division, Floating point numbers and operations.			
Unit-3:	Basic Processing Unit		
Fundamental concepts, Execution of a complete instruction, Multiple bus organization, Hardwired control, Micro programmed control			
Unit-4:	Advance Control unit Design techniques		
Pipelining, Basic concepts, Data hazards Instruction hazards, Influence on Instruction sets, Data path and control consideration Superscalar operation.			
Unit-5:	Memory System		
Basic concepts, Semiconductor RAMs, ROMs , Speed, size and cost, Cache memories Performance consideration, Virtual memory, Memory Management requirements, Secondary storage.			
Unit-6:	I/O Organization		
Accessing I/O devices, Interrupts, Direct Memory Access, Buses, Interface circuits, Standard I/O Interfaces (PCI, SCSI, USB).			
Text Books:			
1.	Computer Organization - Carl Hamacher, ZvonkoVranesic and SafwatZaky, 5th Edition McGraw- Hill, 2002.		
Reference Books			
1.	Computer Organization and Architecture Designing for Performance- William Stallings, 6th Edition, Pearson Education.		
2.	Computer Organization and Design: The hardware / software interface- David A. Patterson and John L. Hennessy, 2nd Edition, Morgan Kaufmann Press.		
3.	Computer Architecture and Organization- John P. Hayes, 3rd Edition, McGraw-Hill		

Course Code:	NCS-102	Course Name: Introduction to Databases	Credits: 4

Course Objectives:

1. To understand the features of Relational database.
2. To describe data models and schemas in DBMS.
3. To use SQL- the standard language of relational databases for database operations.
4. To understand the functional dependencies and design of the databases.

Course Outcome:

1. To study the basic concepts of relational databases
2. Learn and practice data modelling using the entity-relationship and developing database designs.
3. Understand the use of Structured Query Language (SQL) and learn SQL syntax for writing queries.
4. Apply normalization techniques to normalize the databases.

Unit-1: Introduction

Problems in Traditional file oriented approach, Three level architecture of DBMS, basic database components like schema, views, instances, General Architecture of DBMS, Roles of DBA, Data Dictionary, Advantages and Disadvantages of DBMS.

Unit-2: DATA Models

Concepts of Abstraction and Data Model, Discussions on data modeling using Entity Relationship model, Discussions on data modeling using Relational Model, E-R to Relational Conversion.

Unit-3: Relational Algebra

Basics of Relational Algebra, selection, projection, division, cross product Operators Set Operators, Join and its types, writing Relational Algebra notations for user queries.

Unit-4: Basic Normalization

Introduction to attributes, Keys, relationships and their types, Anomalies in databases, understanding Functional Dependencies(Determinant, partial, full, transitive, multi valued, etc), normalization process, First Normal form, Second Normal Form, Third Normal Form etc.

Unit-5: Advance Normalization

Boyce-Codd Normal Form, Fourth Normal Form, Fifth Normal Form.

Unit-6: SQL

Introduction to data retrieval languages like QBE, QUEL, SQL Discussions on SQL, Table , View Definitions ,DDL Statements, DML Statements, DCL Statements , TCL statements , SQL Functions ,Introduction to PL/SQL , Cursors.

Text Books:

1.	Database Management Systems- Raghu Ramakrishnan, Johannes, Gehrke, Tata McGraw Hill.
2	Database System Concepts- Silber Schatz Korth, Tata McGraw Hill.

Reference Books

1.	Fundamental of Database System- Sham Kanth B. Navathe, Pearson Education.
2.	Introduction to Database management System- Bipin Desai, Galgotia Publications.
3.	Oracle Development Language Oracle PL/SQL Programming, Steven Feuerstein , O’Reilly
4.	ORACLE documentations on ORACLE PRESS / Internet.

Course Code:	NCS-103	**Course Name:** Mathematical Foundations for Computer Science	**Credits: 4**

Course Objectives:

Cultivate clear thinking and creative problem solving. Thoroughly train in the construction and understanding of mathematical proofs. Exercise common mathematical arguments and proof strategies.

Course Outcome:

At the end of the course student will be able to Understand the notion of mathematical thinking, mathematical proofs and to apply them in problem solving. Ability to understand use of functions, graphs and their use in programming applications. Apply discrete structures into computing problems, formal specification, artificial intelligence, cryptography, Data Analysis.

Unit-1:

Sets, Venn diagrams, Operations on Sets, Laws of set theory, Power set and Products, Partitions of sets, The Principle of Inclusion and Exclusion.

Unit-2:

Propositions and logical operations, Truth tables , Equivalence, Implications ,Laws of logic, Normal Forms, Predicates and Quantifiers, Mathematical Induction.

Unit-3:

Relations, Paths and Digraphs, Properties and types of binary relations , Operations on relations, Closures, Warshall‟s algorithm, Equivalence and partial ordered relations, Poset, Hasse diagram and Lattice ,Functions: Types of functions - Injective, Surjective and Bijective Composition of functions , Identity and Inverse function, Pigeon-hole principle.

Unit-4:

Permutations, Combinations, Elements of Probability, Discrete Probability and Conditional Probability, Generating Functions and Recurrence Relations, Recursive Functions, Introduction to Functional Programming.

Unit-5:

Graphs Definitions, Paths and circuits: Eulerian and Hamiltonian, Types of graphs, Sub Graphs Isomorphism of graphs.

Unit-6:

Algebraic structures with one binary operation: semigroup, monoid and group, Abelian group Isomorphism, Homomorphism and Automorphism, Cyclic groups, Normal subgroups, Codes and group codes.

Text Books:

1. Discrete Mathematical Structures- Bernad Kolman, Robert Busby, Pearson Education.

2. Discrete Mathematical Structures- C. L. Liu, Second Edition, McGraw-Hill Book

3. Discrete Mathematics and applications- K. H. Rosen, Tata McGraw Hill publishing

Reference Books

1. Discrete Mathematical Structures- Y N Singh, Wiley-India Press.

2. Discrete Mathematics for Computer Scientists and Mathematicians- J. L. Mott, A.Kandel, Prentice Hall of India.

3. Discrete Mathematical Structures with Applications to Computer Science- Discrete Mathematics for Computer Scientists and Mathematicians, Tata Mcgraw-Hill.

Course Code:	NCS-104 A	**Course Name:** Programming Language Concepts	**Credits: 4**
Course Objectives:			
1.To help the students understand the fundamental concepts of programming Languages. 2.To prepare students about the need and use of data structures 3.To prepare students to identify and apply data structures for problem solving			
Course Outcome:			
Understanding the concepts of evolution of programming languages. Understanding the concepts of object oriented languages, functional and logical programming languages Analyzing the methods and tools to define syntax and semantics of a languages Analyzing the design issues involved in various constructs of programming languages Apply the concepts and identify the issues involved in other advanced features of programming languages			
Unit-1:	The role of Programming Languages		
Introduction to Languages, Basic types of languages (Machine, Assembly, High level Languages), Toward Higher-Level Languages, Programming Paradigms, Language Implementation: Bridge the Gap.			
Unit-2:	Language Description: Syntactic Structure		
Expression Notations, Abstract Syntax Trees, Lexical Syntax: Tokens and Spellings, Context-Free Grammars, Grammars for Expressions, Handling Associativity and Precedence.			
Unit-3:	Statements: Structured Programming		
Need for Structured Programming, Syntax-directed Control Flow (conditional, Looping Construct, for, Selection Case) Design considerations: Syntax, Programming with Invariants.			
Unit-4:	Types: Data Representation		
The role of types, Basic types, Arrays: Sequence of elements, Records: Name Fields, Union and Variant Records, Sets, Pointers.			
Unit-5:	Procedure Activations		
Introduction to Procedures, Parameter-Passing Methods, Scope Rules for Names, Nested Scopes in the Source Text, Activation Records, Lexical Scope.			
Unit-6:	Logic Programming		
Computing with relations, Introduction to Prolog, Data Structure in Prolog, Programming Techniques, Control in Prolog, cuts.			
Text Books:			
1.	Programming Languages Concepts and constructs- Ravi Sethi, Pearson Education.		
2.	Concepts of Programming Languages- Robert .W. Sebesta, Pearson Education.		
3.	Programming Language Design Concepts- D. A. Watt, Wiley Dream Tech.		
Reference Books			
1.	Programming Languages- A. B. Tucker, R. E. Noonan, 2nd Edition, TMH.		
2.	Programming Languages- K. C. Louden, 2nd Edition, Thomson Press.		

Course Code:	NCS-104 B	**Course Name:** Object Oriented Programming	**Credits: 4**

Course Objectives:

1. To understanding the principles of object oriented programming
2. To introduce the object oriented way of problem solving.
3.To gain familiarity with the syntax, class hierarchy, environment and simple application construction for an object-oriented programming language

Course Outcome:

1. Acquire a full Object Oriented perspective for analyzing, defining, implementing and evaluating real world problems.
2.Apply and use the object oriented concepts/ techniques, tools in modeling computer based/ software system
3. An ability to apply mathematical foundations, algorithmic principles and computer science theory in the modelling and design of computer-based systems

Unit-1: Introduction

Concept, Benefits and Application of OOP, Structure of C++ Programming, Tokens, expressions and control structures keywords, identifiers, data types and operators in C++.

Unit-2: Functions in C++

Main Function, Function Prototyping, Call by reference, Return by reference, Inline Functions, Default arguments, Function overloading, Friend and Virtual functions.

Unit-3: Class and Objects

Classes, Specifying a class, Defining Member Functions, Making outside function inline, Nesting of Member Functions, private member functions, Arrays within a class, Friend classes, Static class members, Nested classes, Memory allocation for objects, Array to objects, Objects as function arguments.

Unit-4: Constructors and Destructors

Constructors, Parameterized constructors, Multiple constructors in a class, constructors with default arguments, Dynamic initialization of objects, Copy constructors, dynamic constructors, Destructors.

Unit-5: Operator overloading and Type conversion

Defining operator overloading, overloading unary operators, overloading binary operators, overloading binary operators using friends, Manipulation of Strings using operators, Type conversions.

Unit-6: Inheritance, Pointers, Virtual functions and Polymorphism

Single, Multilevel multiple, hierarchical and hybrid inheritance, Virtual base classes, Abstract classes, Pointer to objects, pointer to derived class.

Text Books:

1.	Object Oriented Programming with C++ - E. Balaguruswamy, Tata McGraw Hills.
2.	C++ The Complete Reference– Herbert shield, Tata McGraw Hill Publication.
3.	Object Oriented Programming in C++ - Saurav Sahay, Oxford University Press.

Reference Books

1.	Serial communication-A C++ developers guide – Nelson, M&T Press.
2.	A Treatise on Object Oriented Prog. Using C++ - B. Chandra, Narosa Publishing House.
3.	Object Oriented Programming in C++ - R Rajaram New Age International Publishers.

Course Code:	NCS-105	**Course Name:** Lab-1:DBMS	**Credits: 2**

Course Objectives:

This course aims at giving adequate exposure to students on the Database design and E-R modelling. The course also facilitates students with hands on training on SQL and programming language extension to SQL within the RDBMS environment.

Course Outcome:

1: Model Entity Relationship with E-R diagrams
2: Design database schema considering normalization and relationships within database
3: Write SQL queries to user specifications
4: Develop triggers, procedures, user defined functions and design accurate and PLSQL programs in Oracle and DB2.
5: Use the database from a front end application
6: Prepare technical report on the observations of the experiments

1. Creating database objects
2. Modifying database objects
3. Manipulating the data
4. Retrieving the data from the database server
5. Performing database operations in a procedural manner using pl/sql
6. Performing database operations (create, update, modify, retrieve, etc.,) using front-end tools
7. Design and Develop applications like banking, reservation system, etc.,
8. To create a DDL to perform creation of table, alter, modify and drop column.
9. To create a view for the purpose of display in order to hide the data.
10. Study of DCL extensive feature in order to safeguard information stored in its tables from unauthorized viewing and damage. The rights that allow the user of some or all oracle resources on the server are called privileges.
11. To create a single row functions.
12. Study of PL/SQL features
13. To Perform Banking Operations Using Procedures
14. To carryout payroll application using procedures
15. To write an algorithm to perform database connectivity using MS Access.

Course Code:	NCS-106	**Course Name:** Lab-2: OOP	**Credits: 2**

Course Objectives:

Justify the philosophy of object-oriented design and the concepts of encapsulation, abstraction, inheritance, and polymorphism.
Design, implement, test, and debug simple programs in an object oriented programming language.
Describe how the class mechanism supports Inheritance, Polymorphism

Course Outcome:

Develop program to illustrate basic concept of OOP features and C++ concept
Create and implement program using unary and binary operator overloading
Write program to implement concept of inheritance and polymorphism
Create program to implement concept of abstract class and virtual functions
Develop program using console I/O and file I/O
Develop and implement program using exception handling and templates

1. Write a program to enter mark of 6 different subjects and find out the total mark Write a function using reference variables as arguments to swap the values of pair of integers.
2. Write a inline function to find largest of three numbers.
3. Write a program to find the factorial of a number using recursion.
4. Define a class to represent a bank account which includes the following members as Data members: a) Name of the depositor b)Account Number c)Withdrawal amount d)Balance amount in the account Member Functions: a) To assign initial values b)To deposit an amount c) To withdraw an amount after checking the balance d) To display name and balance.
5. Write the above program for handling n number of account holders using array of objects.
6. Write a C++ program to compute area of right angle triangle, equilateral triangle, isosceles triangle using function overloading concept.
7. Write a C++ program to swap the values two integer members of different classes using friend function.
8. Define a class string and overload to compare two strings and + operator for concatenation of two strings.
9. Write a program for overloading of Unary ++ operator.
10. Define two classes polar and rectangle to represent points in the polar and rectangle systems. Use conversion routines to convert from one system to the other.
11. Write a C++ program to perform matrix addition using operator overloading concept.
12. Consider an example of declaring the examination result. Design three classes student, exam and result. The student has data members such as rollno, name. Create the lass exam by inheriting the student class. The exam class adds data members representing the marks scored in 5 subjects. Derive the result from exam-class and it has own data members like total, avg. write the interactive program into model this relationship.
13. Create a base class called shape, Use this class to store two double type values that could be used to compute the area of figures. Derive two specific classes called triangle and rectangle from the base shape. Add tp the base class, a member function getdata() to initialize base class data members and another member function display_area() to compute and display area of figures. Make display_area() as a virtual function and redefine the function in the derived class to suit their requirements.
14. Using these three classes, design a program that will acdept dimensions of a triangle or a rectangle interactively and display area.
15. Write an interactive program to compute square root of a number. The input value must be tested for validity. If it is negative, the user defined function my_sqrt() should raise an exception.

Course Code:	NCS-107A	**Course Name:** University recognized MOOC (NPTEL / SWAYAM / others) OR Intra / Inter Departmental OR Intra / Inter School	**Credits: 4**
Course Objectives:			
Course Outcome:			

Course Code:	NCS-107 B	**Course Name:** Information Communication Technology (ICT)	**Credits: 4**

Course Objectives:

The ICT curriculum broadly attempts to equip students with an ability to negotiate a range of devices, tools, application, information and resources. The course is offered in chunks of three periods in a week, which include one teacher led session and two hands on sessions. The teacher led session aims to demonstrate techniques and processes and prevent a context to the learning. Following this, students engage themselves with activities which are designed to provide adequate hands on experience.

Course Outcome:

- Create digital art and textual materials
- Use e-resources for learning of curricular subjects
- Interact with ICT devices confidently
- Practise safe, legal and ethical means of using ICT
- Develop digital literacy skills that will enable them to function as discerning students in an increasingly digital society

Unit-1: Computer Fundaments

Information processing Cycle, Brief History of Computers, Environmental, social and ethical issues, laws of ICT, Organizations of Computer System, Introduction to Operating Systems, measures to protect data and systems.

Unit-2: Application Software

Advance Word Processing, working with styles, templates, forms, Advance Spreadsheets, working with multiple worksheets and workbooks, advance functions, Making presentations, working with multimedia presentations.

Unit-3: Database

Working with data, tables, and relationships, creating and customising queries, customising forms, creating reports.

Unit-4: Information and Communication systems

Information systems, networking concepts, functions of network devices, cabling standards, firewall, networking protocols, network security.

Unit-5: Internet Terminology

Web Fundamentals, Web Security, Communication protocols, creating web pages, working with images and multimedia, working with special effects.

Unit-6: Applications of ICT

Career options in ICT, Job search on Internet and other media, Formatting Resume or CVs, Formatting Application Letters, working with publications.

Text Books:

1. Computer Fundamentals, Ms Office and Internet and Web Technology- Dinesh Maidasani, Firewall Media.

2. Computer Fundamentals- Anita Goel, Person Education.

3. Computer Fundamentals- P. K. Sinha, EduTech Learners.

Reference Books

1. MS Office for Dummies- Wallace Wang, Wiley Publishing, Inc.

2. Ms Office Step by Step- John Lambert, Microsoft Press.

Course Code:	NCS-108	**Course Name:** SK-01	**Credits:** 1
Course Objectives:			
Course Outcome:			

Course Code:	NCS-201	**Course Name:** Operating System Concepts	**Credits: 4**

Course Objectives:

1. To learn the fundamentals of Operating Systems.
2. To learn the mechanisms of OS to handle processes and threads and their communication
3. To learn the mechanisms involved in memory management in contemporary OS
4. To gain knowledge on distributed operating system concepts that includes architecture, Mutual exclusion algorithms, deadlock detection algorithms and agreement protocols
5. To know the components and management aspects of concurrency management
6. To learn programmatically to implement simple OS mechanisms

Course Outcome:

- Analyse the structure of OS and basic architectural components involved in OS design
- Analyse and design the applications to run in parallel either using process or thread models of different OS
- Analyse the various device and resource management techniques for timesharing and distributed systems

Unit-1: Overview of Operating System

Operating system objectives and functions, Evolution of OS, Characteristics of modern OS, Basic concepts: Processes, Files, System calls, Shell, Kernel architectures: Monolithic, Micro-kernel, Layered, Kernel mode of operations.

Unit-2: Process Management

Process description: Process, Process States, Process Control Block (PCB), Threads, Thread management. Process Scheduling: Types, Comparison of different scheduling policies.

Unit-3: Process Co-ordination

Principles of Concurrency, Race condition and critical section, Mutual Exclusion, Semaphores, Message Passing, Deadlock: Principles of Deadlock, Deadlock Detection, Deadlock Avoidance, Deadlock Prevention.

Unit-4: Memory Management

Memory Management Requirements, Memory Partitioning, Virtual memory: Paging; Segmentation; Page replacement policies.

Unit-5: File System

File concept, Access methods, Directory and disk structure, File system mounting, File sharing, Protection.

Unit-6: Input Output Management

I/O Devices, Organization of the I/O Function, Operating System Design Issues, I/O Buffering, Disk Scheduling and disk scheduling algorithms.

Text Books:

1.	Operating Systems Concepts- Silberschatz A., Galvin P., Gagne G, Wiley Publication.
2.	Modern Operating Systems, Andrew S. Tanenbaum, III rd Edition, PHI Publication.

Reference Books

1.	Operating System-Internal and Design Principles, William Stallings, Pearson Education.
2.	Principles of Operating Systems-Naresh Chauhan, First Edition, Oxford University press.
3.	Operating Systems in Depth- Thomas W. Doeppner, Wiley Publications..

Course Code:	NCS-202	**Course Name:** Elementary Data Structures and Algorithms	**Credits: 4**

Course Objectives:

- IT will demonstrate familiarity with major algorithms and data structures.
- Analyse performance of algorithms.
- Choose the appropriate data structure and algorithm design method for a specified application.
- Use various data structures effectively in application programs.
- Demonstrate understanding of various sorting algorithms, including bubble sort, insertion sort, selection sort, merge sort, quick sort and heap sort.

Course Outcome:

- Explain the organization and operations of data structures Stack, Queues, Trees, Graphs, Heaps and Hash tables.
- Compare and contrast the functionalities and applications of different data structures
- Demonstrate specific search and sort algorithms using data structures given specific user requirements.
- Apply the operations of data structures in designing software procedures based on specific

Unit-1: Introduction to Algorithm

Data, Variables (Local and Global), Data types, arrays Introduction to Algorithm, The efficiency of Algorithms, Analysis of Algorithms, overview of Space and Time Complexities, some fundamental algorithms for exchange , counting , summation .

Unit-2: Introduction to data structures

Introduction to data structures, Basic terminology, Primitive data structure operations
Overview of STACKS, QUEUES, LINKED LISTS, BINARY TREES and GRAPHS (Basic Definition , Representations, Characteristics , Types, Applications)

Unit-3: Tree and Graph

Minimum Spanning Trees, Growing a minimum spanning tree, The algorithms of Kruskal and Prim Graphs: DFS and BFS algorithms associated with Graphs, Single-source shortest Paths, The Bellman-ford algorithm.

Unit-4: Sorting and Searching

Introduction to searching and sorting problems, Linear search , Binary search, Selection sort , Bubble sort , Insertion sort , Merge sort, Complexities of searching and sorting algorithms.

Unit-5: Divide and Conquer Techniques

Divide and conquer, General method, Binary search, Merge sort, Strassen’s matrix multiplication.

Unit-6: Advanced Data Structure

Introduction to Greedy method, The general method, Container loading knapsack problem,
Introduction to Dynamic Programming, General method, Introduction to NP Theory.

Text Books:

1.	Fundamentals of Computer Algorithms- Ellis Horowitz, Satraj Sahani,

Reference Books

1.	Data Structures, Lipschutz , Tata McGraw Hills.
2.	How to solve it by Computers- R.G. Dromey , 8th Edition , Pearson Education.

Course Code:	NCS-203	**Course Name:** Programming in Java	**Credits: 4**

Course Objectives:

The objective of this course is to create Java programs that leverage the object-oriented features of the Java language, such as encapsulation, inheritance and polymorphism, use data types, arrays and other data collections, implement error-handling techniques using exception handling, create and event-driven GUI using Applet.

Course Outcome:

To design, write, compile, test and execute straightforward programs using a high level language.
To implement, compile, test and run Java programs comprising more than one class, to address a particular software problem
To demonstrate the ability to use simple data structures like arrays in a Java program.
To demonstrate the ability to employ various types of selection constructs in a Java program.
To employ a hierarchy of Java classes to provide a solution to a given set of requirements.

Unit-1: Introduction to Java

History, Features, How java differ from C and C++?, Java program structure, Java tokens, Java Statements, Java virtual machine, Command line arguments, Constants, Variable, Data types, Type casting.

Unit-2: Operators and Expression

Decision making and branching, Decision making and looping, Class, Methods, Objects, Constructors, Method overloading, Static members, nesting of methods.

Unit-3: Inheritance

Overriding methods, Final variables, Final methods, Final Classes, Finalizer method, Abstract methods, Abstract Classes, Visibility Control, Interface, Arrays, Strings, Vectors, Wrapper Classes.

Unit-4: System Packages

Naming conventions, Creating and accessing packages, Introduction to multithreaded programming, Creating and extending threads, Life cycle of thread, Thread exception, Thread priority, Synchronization, Implementing Runnable interface, Types of errors, Exceptions, Exception handling code, Multiple catch statements, finally statement, Throwing our own exceptions, Exception for debugging.

Unit-5: Introduction to Applet

How applet differ from application?, Applet code, Applet life cycle, Creating an executable applet, designing a web page, Applet tag, Passing parameter to applet.

Unit-6: The Graphic Class

Lines, Rectangles, Circles, Ellipses, Arcs, Polygons, Line graphs, Bar charts, Control loops in applet.

Text Books:

1.	Programming with Java A Primer – E.Balaguruswamy, McGraw Hill.
2.	Java 7 Programming Black Book -Kogent Learning Solutions Inc,DreamTech press.

Reference Books

1.	Java Fundamentals A comprehensive introduction- Herbert Schildt, Dale Skrien, McGraw Hill Education.
2.	The Complete Reference, Java 2 -, Herbert Schild, (Fourth Edition) - TMH.
3.	Core Java Volume-I Fundamentals- Horstmannand Cornell, - Pearson Education.

Course Code:	NCS-204 A	**Course Name:** System Analysis and Design	**Credits: 4**
Course Objectives:			
System analysis helps in discovering means to design systems where sub-system may have apparently conflicting objectives. It helps in achieving inter compatibility and unity of purpose of sub-systems. It offers a means to create understanding of the complex structures			
Course Outcome:			
After successfully completing this course, students will understand concepts of Analysis and Designing Information Systems. Students will understand writing system proposals, system development scheduling, and cost-benefits analysis etc. also dealing with quality assurance.			
1: To learn basic things of systems, System development Life cycle, and System Analyst. 2: To determine specific needs of system. 3: Discuss approaches and tasks of system. Planning for developing system 4: Evaluate tools and techniques.			
Unit-1:	Introduction		
System Definition, Characteristics, Elements and Types of system, Need of System Analysis and design, Role and Qualities of System Analyst, System Development Life Cycle.			
Unit-2:	Feasibility Study		
Project Initiation, Feasibility study, Ascertaining HW/SW needs, Criteria for HW/SW selection, Make v/s Buy Decision, Cost Benefit Analysis.			
Unit-3:	Decision Modules		
Structured Analysis tools- DFD, Data Dictionary, Decision Tree, Decision Table, Structured English, Activity planning control, Activity Diagrams, Case modeling, UML, Class Diagram.			
Unit-4:	Scheduling		
System Proposal, Project Scheduling, Information Gathering Tools- Interviews, Questionnaire, JAD, Prototyping.			
Unit-5:	Design		
System Design, Input/output Design, From Design, Database Design, File organization.			
Unit-6:	Implementation		
System Implementation Plan, Activity Network for Conversion, Combating Resistance to Change, System Testing, Test Plan AND test data, Types of System Test, Quality Assurance, Documentation.			
Text Books:			
1.	System Analysis and Design- Kendall and Kendall, Pearson Education, Inc.		
Reference Books			
1.	System Analysis and Design- E. M. Awad, Galgotia Publications Pvt. Ltd		
2.	Modern System Analysis and Design - Jeffrey A. Hoffer, Prentice-Hall, Inc		

Course Code:	NCS-204 B	**Course Name:** Data Communications	**Credits: 4**
Course Objectives:			
At the end of the course, students will be able to understand basic computer network technology. Understand and explain various components of computer networks. Identify the different types of network topologies and protocols. Enumerate the layers of the OSI model and TCP/IP. Explain the function(s) of each layer. Identify the different types of network devices and their functions within a network. Understand and build the skills of routing mechanisms.			
Course Outcome:			
1: Describe the building blocks of Computer Networks 2: Explain the functionalities and protocols of various layers in ISO/OSI Network model. 3: Implement a suitable routing strategies for a given network 4: Use suitable transport/application layer protocol based on application requirements			
Unit-1:	Introduction		
Introduction to Communications Model, Data Communications, Networks, The Internet, An Example Configuration, The Need for a Protocol Architecture, The TCP/IP Protocol Architecture, The OSI Model, Standardization Within a Protocol Architecture, Traditional Internet-Based Applications, Multimedia.			
Unit-2:	Data Transmission		
Data transmission, Concepts and Terminology, Analog and Digital Data Transmission Transmission Impairments, Channel Capacity.			
Unit-3:	Transmission Media		
Transmission media, Guided Transmission Media, Wireless Transmission, Wireless Propagation, Line-of-Sight Transmission.			
Unit-4:	Digital Data Communication Techniques		
Digital Data communication techniques, Asynchronous and Synchronous Transmission, Types of Errors, Error Detection, Error Correction, Line Configurations.			
Unit-5:	Data Link Control Protocols		
Data link Control protocols, Flow Control, Error Control, High-Level Data Link Control (HDLC).			
Unit-6:	Multiplexing		
Frequency Division Multiplexing, Synchronous Time Division Multiplexing, Statistical Time Division Multiplexing, Asymmetric Digital Subscriber Line,xDS.			
Text Books:			
1.	Data and Computer Communications- William Stallings, 8th Edition Pearson		
Reference Books			
1.	Data Communications and Networking, Behrouz A. Forouzan, 2nd Edition, McGraw Hill Publication.		

Course Code:	NCS-205	**Course Name:** Lab-3: Data Structures	**Credits: 2**

Course Objectives:

- To develop skills to design and analyse simple linear and nonlinear data structures
- To strengthen the ability to identify and apply the suitable data structure for the given real world problem
- To gain knowledge in practical applications of data structures

Course Outcome:

- To learn elementary data structures such as stacks, queues, linked lists, trees and graphs
- To design and analyze the time and space efficiency of the data structure
- To identity the appropriate data structure for given problem
- To have practical knowledge on the application of data structures
- To discuss different data structures to represent real world problems
- To design algorithms to solve the problems.

1. Array implementation of List Abstract Data Type (ADT)
2. Linked list implementation of list ADT
3. Array implementations of stack ADT
4. Linked list implementations of stack ADT

The following three exercises are to be done by implementing the following source files

a) Program for ‘Balanced parenthesis’
b) Array implementation of stack ADT
c) Linked list implementation of stack ADT
d) Program for ‘Evaluating Postfix Expressions’

An appropriate header file for the stack ADT should be # included in (a) and (d)

5. Implement the application for checking ‘balanced parenthesis’ using array Implementation of stack ADT (by implementing files (a) and (b) given above)
6. Implement the application for checking ‘Balanced Parenthesis’ using linked list Implementation of stack ADT (by using file (a) from experiment 6 and implementing file (c))
7. Implement the application for ‘Evaluating Postfix Expressions’ using array and linked list implementations of Stack ADT (by implementing file (d) and using file (b), and then by using files (d) and (c))
8. Queue ADT
9. Search Tree ADT – Binary Search Tree
10. Heap Sort, Quick Sort

Lecture: 0; Practical: 45; Total: 45

<table>
<tr><td><b>Course Code:</b></td><td>NCS-206</td><td><b>Course Name:</b> Lab-4: Java Programming</td><td><b>Credits: 2</b></td></tr>
<tr><td colspan="4"></td></tr>
<tr><td colspan="4"><b>Course Objectives:</b></td></tr>
<tr><td colspan="4">To enable the students practice the concepts of java programming language and develop solutions for real world problems.</td></tr>
<tr><td colspan="4"></td></tr>
<tr><td colspan="4"><b>Course Outcome:</b></td></tr>
<tr><td colspan="4">1: Understand the enabling technologies for building internet applications. Understand
2: Write Java programs for techniques and features of the networking and remote method development to Construct a internet application. Apply
3: Implement packages, access specifiers and interfaces in a program Apply
4: Implement Program for Events and interactivity using Layout Manager. Apply
5: Generate program for network chatting analyse
6: Write technical report on the observations from the experiments</td></tr>
<tr><td colspan="4"></td></tr>
<tr><td colspan="4">1. Use of Objects
2. Using classes and inheritance
3. JNI concepts
4. Multithread applications
5. Exception handling
6. Implementing packages, access specifiers and interfaces
7. Streams
8. JDBC program using different statements
9. Applet program for Animation text, images and sounds
10. Events and interactivity using Layout Manager.
11. Socket program for network chatting
12. Client server application using RMI techniques</td></tr>
<tr><td></td><td colspan="2"></td><td></td></tr>
</table>

Course Code:	NCS-207A	**Course Name:** University recognized MOOC (NPTEL / SWAYAM / others) OR Intra / Inter Departmental OR Intra / Inter School	**Credits:4**
Course Objectives:			
Course Outcome:			

Course Code:	NCS-207 B	**Course Name:** Introduction to Linux	**Credits: 4**
Course Objectives:			
To enable the students practice the concepts of Operating systems and develop solutions for real world problems.			
Course Outcome:			
1: Have a good orientation towards concept-based approach and practical-based approach 2: Students will be able to describe the components of a modern operating system 3: Apply operating system concepts practically 4: Apply the concepts of operating systems design to practical problems			
Unit-1:	Synaptic Package Manager		
Synaptic Package Manager, How to install packages, Basic Commands, Commands with example, Command interpreter, Shell, Using man, Apropos.			
Unit-2:	General Purpose Utilities in Linux		
Echo, uname, who, passwd, date, cal, Brief overview on Files and directories, pwd, ls, cat, File System, File, Directory, File Inode, Types of Files,Home directory and Current directory, Change Directory(cd), mkdir,rmdir, cat, rm, cp, mv, cmp, wc, File Attributes.			
Unit-3:	Redirection Pipes		
Input, output and error stream, Redirection : > and >>, Working with Linux Process, Process, Shell process, Process spawning - parent and child process, Process attributes - pid, ppid, Init Process, User process and System process, ps with options.			
Unit-4:	The Linux Environment		
Environment variable vs Local variables, set command, env command, SHELL, HOME, PATH, LOGNAME, PS1, PS2, history, ! and ~, alias, Basics of System Administration, Root login-su, User management - UID, GID, useradd, usermod, userdel, Discs – Du, df, Simple filters, Head, tail, sort, cut, paste.			
Unit-5:	The grep command		
To see the content of a file, To list the entries of a particular stream, To ignore cases, Lines that do not match the pattern, To list the line numbers, To store the result in another file, To know the count, To match more than one pattern, Character class, To match a pattern at the end of the file, The sed command, Line Addressing, Context Addressing, Basics of awk, Awk Preliminaries, Selection criteria, action, Fields, Regular expressions, NR - number of records, Variables.			
Unit-6:	Networking tools		
Ping, Telnet, ftp, ssh, scp and sftp, Linux Process, Fork, Exec, Wait, Nice, Kill with options, More about Linux Process, Cron, crontab.			
Text Books:			
1.	**Fedora 10 and Red Hat Enterprise Linux Bible- Christopher Negus, Wiley Publishing**		
2.	**Linux For Dummies-** Dee-Ann LeBlanc, R. K. Blum, **Wiley Publishing.**		
Reference Books			
1.	**Ubuntu for Non-Geeks, 2nd Edition: A Pain-Free, Project-Based, Guide book- Rickford Grant, Phil Bull, William Pollock Press.**		

Course Code:	NCS-208	**Course Name:** SK-02	**Credits: 1**
Course Objectives:			
Course Outcome:			

Director
School of Computational Sciences
S.R.T.M.University, Nanded (M.S.)