ACADEMIC (1-BOARD OF STUDIES) SECTION

Phone: (02462) 229542  
Fax: (02462) 229574  
Website: www.srtmun.ac.in  
E-mail: bos.srtmun@gmail.com

परिषद क्र

या परिषदनावे सर्व संघीतांत जनविभाग येदे को, दिनंक ०८ जून २०१९ रेची संपन 
झट्टिया ४४१८ मा. विधा परिषद् बैठकीतील ऐनेवडचा विषय ₹१९/४४–२०१९ चा उद्घाटनादर प्रस्तुत 
विद्यापीठात वर्षाचे बाळांसह C.B.C.S. (Choice Based Credit System) Pattern पुसारे अभ्यासक्रम 
शैक्षणिक वर्ष २०१९–२० पातून लागू करणावावर.

1. Botany
2. Certificate Course in Industrial Safety, Health and Environmental Management (SHM)
3. Chemistry
4. Computer Application
5. Computer Network
6. Computer Science
7. Geophysics
8. Mathematics
9. M.C.A.
10. Microbiology
11. Physics
12. Zoology

सदरील व प्रस्तुत अभ्यासक्रम विद्यापीठातन या संकेतस्थळवर

उपलब्ध आहेत. तरी सदरील बाबी ही सर्व संघीतांत निर्धाराच आणून आहे.

शास्ती 'परिषद,'
विनामुळे, नांदूडे — ४३९ ६०६.
त्यांक: २०१९-२० /संख्या/संख्या /संख्या/संख्या अभ्यासक्रम/२०१९–२०/४६५

दिनंक: ११.०७.२०१९.
प्रमाणीती व पुमेरल कार्यवाहीस्थळ:

1) म. कृष्णसम्पादनाचे कार्यालय, प्रस्तुत विद्यापीठ.
2) म. संसाधन, पारिश्रमिक व मूल्यमापन मंडळ यांचे कार्यालय, प्रस्तुत विद्यापीठ.
3) म. संसाधन, सर्व संघीत संघीत, विद्यापीठ परिषद, प्रस्तुत विद्यापीठ.
4) साहित्यक कृतिविविध, पदव्युतर विभाग, प्रस्तुत विद्यापीठ.
5) उपकृतिविविध, पात्रता विभाग, प्रस्तुत विद्यापीठ.
6) सिस्टेम एंटरप्राइज, शैक्षणिक विभाग, प्रस्तुत विद्यापीठ.
Swami Ramanand Teerth Marathwada University, Nanded
(NAAC Re-accredited with ‘A’ Grade)

Syllabus of

M.Sc. (Computer Science) (Campus)
(2 years) (Revised CBCS pattern)

Introduced from Academic Year 2019-2020

Program code: SCS-S-MCS-PG (13-2-2-01)
M.Sc. Computer Science (Campus)

M.Sc. Computer Science (2years) program / degree is a specialized program in latest advances in computer science issues. It builds the student on higher studies and research awareness in overall computational, IT and ICT fields so as to become competent in the current race and development of new computational sciences. The duration of the study is of four semesters, which is normally completed in two years.

CBCS pattern

The M.Sc. Computer Science program as per CBCS (Choice based credit system) pattern, in which choices are given to the students under open electives and subject electives. The students can choose open electives from the wide range of options to them.

Eligibility and Fees

The eligibility of a candidate to take admission to M.Sc. Computer Science program is as per the eligibility criteria fixed by the University. More details on admission procedure and fee structure can be seen from the prospectus of the college / institution as well as on website of the University.

Credit Pattern

Every course has corresponding grades marked in the syllabus structure. There are 25 credits per semester. A total of 100 credits are essential to complete this program successfully. The Grading pattern to evaluate the performance of a student is as per the University rules.

Every semester has a combination of Theory (core or elective) courses and Lab courses. Each theory course has 04 credits which are split as 02 external credits and 02 internal credits. The university shall conduct the end semester examination for 02 external credits. For theory internal credit, student has to appear for 02 class test (15 marks) and 01 assignment (20 marks). Every lab course has 02 credits which are split as 01 external credit and 01 internal credit. For lab internal credit, the student has to submit Laboratory Book (05 marks) and remaining 20 marks are for the Lab activities carried out by the student throughout the semester. For lab external credit, 20 marks are reserved for the examinational experiment and 05 marks are for the oral / viva examinations. There is a special skill based activity of 01 internal credits per semester which shall inculcate awareness regarding the domain of computers, IT, and ICT.

The open elective has 04 credits which are purely internal. If students are opting for MOOCs as open elective, then, there must be a Faculty designed as MOOCs course coordinator who shall supervise learning through MOOCs. This is intentionally needed as the MOOCs course coordinator shall verify the MOOC details including its duration, staring date, ending date, syllabus contents, mode of conduction, infrastructure feasibility, and financial feasibility during start of each semester. This is precautionary as the offering of the MOOCs through online platforms are time specific and there must be proper synchronization of semester duration with the MOOCs duration. Students must opt for either institutional / college level open elective or a course from University recognized MOOCs platforms as open electives.

The number of hours needed for completion of theory and practical courses as well as the passing rules, grading patterns, question paper pattern, number of students in practical batches, etc shall be as per the recommendations, norms, guidelines and policies of the UGC, State Government and the SRTM University currently operational. The course structure is supplemented with split up in units and minimum numbers of hours needed for completion of the course, wherever possible.

Under the CBCS pattern, students would graduate M.Sc. Computer Science with a minimum number of required credits which includes compulsory credits from core courses, open electives and program specific elective course. All students have to undergo lab / practical activities leading to specific credits and project development activity as a part of professional UG program.

1. M.Sc. Computer Science Degree / program would be of 100 Credits. Total credits per semester= 25
2. Each semester shall consist of three core courses, one elective course, one open elective course and two practical courses. Four theory courses ( core+elective) = 16 Credits. Two practical / Lab courses= 4 Credits in total (02 credits each) , One Open elective= 4 credit, One skill
3. enhancement activity of 01 credits.
4. One Credit = 25 marks , Two Credits = 50 Marks, Four Credits = 100 Marks
PEO, PO and CO Mappings

1. **Program Name**: M.Sc.(CS) Campus { SCS-S-MCS-PG (13-2-2-01)}

2. **Program Educational Objectives**: After completion of this program, the graduates / students would

   | PEO I : Technical Expertise | Implement fundamental domain knowledge of core courses for developing effective computing solutions by incorporating creativity and logical reasoning. |
   | PEO II : Successful Career | Deliver professional services with updated technologies in computational science based career. |
   | PEO III : Hands on Technology and Professional experience | Develop leadership skills and incorporate ethics, team work with effective communication & time management in the profession. |
   | PEO IV : Interdisciplinary and Life Long Learning | Undergo higher studies, certifications and research programs as per market needs. |

3. **Program Outcome(s)**: Students / graduates will be able to

   - **PO1**: Apply knowledge of mathematics, science and algorithm in solving Computer problems.
   - **PO2**: Generate solutions by understanding underlying computer science environment
   - **PO3**: Design component, or processes to meet the needs within realistic constraints.
   - **PO4**: Identify, formulate, and solve problems using computational temperaments.
   - **PO5**: Comprehend professional and ethical responsibility in computing profession.
   - **PO6**: Express effective communication skills.
   - **PO7**: Recognize the need for interdisciplinary, and an ability to engage in life-long learning.
   - **PO8**: Actual hands on technology to understand it’s working.
   - **PO9**: Knowledge of contemporary issues and emerging developments in computing profession.
   - **PO10**: Utilize the techniques, skills and modern tools, for actual development process
   - **PO11**: Function effectively as an individual and as a member or leader in diverse teams and in multidisciplinary settings in actual development work
   - **PO12**: Research insights and conduct research in computing environment.

4. **Course Outcome(s)**: Every individual course under this program has course objectives and course outcomes (CO). The course objectives rationally match with program educational objectives. The mapping of PEO, PO and CO is as illustrated below

5. **Mapping of PEO& PO and CO**

<table>
<thead>
<tr>
<th>Program Educational Objectives</th>
<th>Thrust Area</th>
<th>Program Outcome</th>
<th>Course Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td>PEO I</td>
<td>Technical Expertise</td>
<td>PO1, PO2, PO3, PO6</td>
<td>All core courses</td>
</tr>
<tr>
<td>PEO II</td>
<td>Successful Career</td>
<td>PO4, PO5, PO11, P12</td>
<td>All discipline specific electives courses</td>
</tr>
<tr>
<td>PEO III</td>
<td>Hands on Technology and Professional experience</td>
<td>PO8, PO10</td>
<td>All Lab courses</td>
</tr>
<tr>
<td>PEO IV</td>
<td>Interdisciplinary and Life Long Learning</td>
<td>PO7, PO9, PO12</td>
<td>All open electives and discipline specific electives</td>
</tr>
</tbody>
</table>

The detailed syllabus is as below,
**CBCS Revised Syllabus w.e.f AY: 2019-2020**

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

{SCS-S-MCS-PG (13-2-2-01)}

---

<table>
<thead>
<tr>
<th>Sr. No</th>
<th>Course category</th>
<th>Course Code</th>
<th>Course Title</th>
<th>Internal credits</th>
<th>External credits</th>
<th>Total credits</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Core Subjects</td>
<td>NCS-101</td>
<td>Computer System Organization</td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NCS-102</td>
<td>Introduction to Databases</td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NCS-103</td>
<td>Mathematical Foundations for Computer Science</td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
</tbody>
</table>

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*
<table>
<thead>
<tr>
<th>Sr. No</th>
<th>Course category</th>
<th>Course Code</th>
<th>Course Title</th>
<th>Internal credits</th>
<th>External credits</th>
<th>Total credits</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Second Semester</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.</td>
<td>Core Subjects</td>
<td>NCS-201</td>
<td>Operating System Concepts</td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>NCS-202</td>
<td>Elementary Data Structures and Algorithms</td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>NCS-203</td>
<td>Programming in Java</td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Choose any one from below elective subjects</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Elective Subject</td>
<td>NCS-204 A</td>
<td>System Analysis and Design</td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NCS-204 B</td>
<td>Data Communications</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Practical /Lab</td>
<td>NCS-205</td>
<td>Lab-3: Data Structures</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NCS-206</td>
<td>Lab-4: Java Programming</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>Open Elective</td>
<td>NCS-207A</td>
<td>University recognized MOOC (NPTEL / SWAYAM / others) OR Intra / Inter Departmental OR Intra / Inter School OR</td>
<td>4</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NCS-207 B</td>
<td>Introduction to Linux</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Skill based Activity</td>
<td>NCS-208</td>
<td>SK-02</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Total credits</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>25</td>
</tr>
<tr>
<td>Sr. No</td>
<td>Course category</td>
<td>Course Code</td>
<td>Course Title</td>
<td>Internal credits</td>
<td>External credits</td>
<td>Total credits</td>
</tr>
<tr>
<td>-------</td>
<td>-----------------</td>
<td>-------------</td>
<td>--------------</td>
<td>------------------</td>
<td>------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>1.</td>
<td>Core Subjects</td>
<td>NCS-301</td>
<td>Windows Programming</td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>NCS-302</td>
<td>Computer Networks</td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>NCS-303</td>
<td>Compiler Designing</td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
</tbody>
</table>

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 |
<table>
<thead>
<tr>
<th>Sr. No</th>
<th>Course category</th>
<th>Course Code</th>
<th>Course Title</th>
<th>Internal credits</th>
<th>External credits</th>
<th>Total credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Core Subjects</td>
<td>NCS-401</td>
<td>Mobile Application Development</td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>NCS-402</td>
<td>Advanced Databases and Administration</td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>NCS-403</td>
<td>Major Project development Activity</td>
<td>0</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Elective Subject</td>
<td>NCS-404 A</td>
<td>Internet of Things (IoT)</td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NCS-404 B</td>
<td>Big Data Analytics</td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Lab / Practical</td>
<td>NCS-405</td>
<td>Lab-7: Mobile Application Development</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NCS-406</td>
<td>Lab-8: Advanced Databases</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>Open Elective</td>
<td>NCS-407A</td>
<td>University recognized MOOC (NPTEL / SWAYAM / others) OR Intra / Inter Departmental OR Intra / Inter School OR</td>
<td>4</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NCS-407 B</td>
<td>Programming in Python</td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>Skill based Activity</td>
<td>NCS-408</td>
<td>SK-04</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

**Total credits**: 25
# Course

<table>
<thead>
<tr>
<th>Course Code:</th>
<th>NCS-101</th>
<th>Course Name:</th>
<th>Computer System Organization</th>
<th>Credits: 4</th>
</tr>
</thead>
</table>

## 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
- Understanding 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:

## Reference Books
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.

<table>
<thead>
<tr>
<th>Unit-1: Introduction</th>
<th>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.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unit-2: DATA Models</td>
<td>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.</td>
</tr>
<tr>
<td>Unit-3: Relational Algebra</td>
<td>Basics of Relational Algebra, selection, projection, division, cross product Operators Set Operators, Join and its types, writing Relational Algebra notations for user queries.</td>
</tr>
<tr>
<td>Unit-4: Basic Normalization</td>
<td>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.</td>
</tr>
<tr>
<td>Unit-5: Advance Normalization</td>
<td>Boyce-Codd Normal Form, Fourth Normal Form, Fifth Normal Form.</td>
</tr>
<tr>
<td>Unit-6: SQL</td>
<td>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.</td>
</tr>
</tbody>
</table>

**Text Books:**


**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:

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:

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

Reference Books
<table>
<thead>
<tr>
<th>Course Code: NCS-104 A</th>
<th>Course Name: Programming Language Concepts</th>
<th>Credits: 4</th>
</tr>
</thead>
</table>

**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 language.
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

**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.

**Reference Books:**
<table>
<thead>
<tr>
<th>Course Code:</th>
<th>NCS-104 B</th>
<th>Elective</th>
<th>Course Name: Object Oriented Programming</th>
<th>Credits: 4</th>
</tr>
</thead>
</table>

**Course Objectives:**
1. To understand 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++

**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:**

**Reference Books**
<table>
<thead>
<tr>
<th>Course Code:</th>
<th>NCS-105</th>
<th>Course Name: Lab-1:DBMS</th>
<th>Credits: 2</th>
</tr>
</thead>
</table>

**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.

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

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.
<table>
<thead>
<tr>
<th>Course Code:</th>
<th>NCS-106</th>
<th>Course Name:</th>
<th>Lab-2: OOP</th>
<th>Credits: 2</th>
</tr>
</thead>
</table>

**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 an 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 class 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 to 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 accept 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.
<table>
<thead>
<tr>
<th>Code:</th>
<th>NCS-107 A</th>
<th>First semester</th>
<th><strong>Open Elective</strong></th>
<th>Credits: 04</th>
</tr>
</thead>
</table>

**Open Elective : University recognized MOOC (NPTEL / SWAYAM / others) OR Intra / Inter Departmental courses**

OR

<table>
<thead>
<tr>
<th>Course Code:</th>
<th>NCS-107 B</th>
<th>Course Name: Information Communication Technology (ICT)</th>
<th>Credits: 4</th>
</tr>
</thead>
</table>

**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.

<table>
<thead>
<tr>
<th>Unit-6:</th>
<th>Applications of ICT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Career options in ICT, Job search on Internet and other media, Formatting Resume or CVs, Formatting Application Letters, working with publications.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Text Books:</th>
</tr>
</thead>
<tbody>
<tr>
<td>2. Computer Fundamentals- Anita Goel, Person Education.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Reference Books</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. MS Office for Dummies- Wallace Wang, Wiley Publishing, Inc.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Code:</th>
<th>First semester</th>
<th>Skill based Activity</th>
<th>Credits:</th>
</tr>
</thead>
<tbody>
<tr>
<td>NCS-108</td>
<td>SK- 01: PC Assembly and Maintenance</td>
<td></td>
<td>01</td>
</tr>
</tbody>
</table>

Scope: Practically understand the PC and surrounding peripherals. The student will assemble / setup and upgrade personal computer systems; install OS and other application software, diagnose and isolate faulty components; optimize system performance and install / connect peripherals.
<table>
<thead>
<tr>
<th>Course Code</th>
<th>NCS-201</th>
<th>Course Name: Operating System Concepts</th>
<th>Credits: 4</th>
</tr>
</thead>
</table>

**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

**Unit-2:** Process Management

**Unit-3:** Process Co-ordination

**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

**Text Books:**

**Reference Books:**
<table>
<thead>
<tr>
<th>Course Code: NCS-202</th>
<th>Course Name: Elementary Data Structures and Algorithms</th>
<th>Credits: 4</th>
</tr>
</thead>
</table>

**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 requirements.

**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

**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

**Text Books:**
1. Fundamentals of Computer Algorithms- Ellis Horowitz, Satraj Sahani,

**Reference Books**
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

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:

Reference Books
2. The Complete Reference, Java 2 - , Herbert Schildt, (Fourth Edition) - TMH.
**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

#### Unit-6: Implementation

### Text Books:

### Reference Books
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

Unit-2: Data Transmission
Data transmission, Concepts and Terminology, Analog and Digital Data Transmission Transmission Impairments, Channel Capacity.

Unit-3: Transmission Media

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:

Reference Books
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>
<thead>
<tr>
<th>Course Code:</th>
<th>NCS-206</th>
<th>Course Name:</th>
<th>Lab-4: Java Programming</th>
<th>Credits:</th>
<th>2</th>
</tr>
</thead>
</table>

**Course Objectives:**
To enable the students practice the concepts of java programming language and develop solutions for real world problems.

**Course Outcome:**
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

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

<table>
<thead>
<tr>
<th>Code:</th>
<th>First semester</th>
<th>Open Elective</th>
<th>Credits:</th>
<th>04</th>
</tr>
</thead>
<tbody>
<tr>
<td>NCS- 207 A</td>
<td></td>
<td>University recognized MOOC  (NPTEL / SWAYAM / others) OR Intra / Inter Departmental courses</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**OR**

<table>
<thead>
<tr>
<th>Course Code:</th>
<th>NCS-207 B</th>
<th>Course Name:</th>
<th>Introduction to Linux</th>
<th>Credits:</th>
<th>4</th>
</tr>
</thead>
</table>

**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

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

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

Code: NCS-208 Second semester Skill based Activity
SK-02: Networking Essentials Credits: 01

Scope: Networking Essentials deals with knowing what is a network, how to install, configure, and troubleshoot a computer network. It includes knowledge of the fundamental building blocks that form a modern network, such as various cables, switches, routers, connectors, LAN-NIC cards and network operating systems. It then provides in-depth coverage of the most important concepts in contemporary networking like connecting computers/ peripherals, servers and clients, Wi-Fi connectivity, etc. Students are expected to have the skills to build a network / LAN from scratch and maintain, upgrade, and troubleshoot an existing network.