# HSNC University Mumbai

(2025-2026)
Ordinances and Regulations
With Respect to
For the Programme Under

# Bachelor of Vocational Studies in Web Technologies

**Curriculum – Second Year B Voc Programme**
**Semester-III and Semester -IV**

**(With effect from the academic year 2025-2026)**

## HSNC UNIVERSITY, MUMBAI

## Board of Studies in BVOC
## In the subject of Web Technologies, KC College.

**1. Name of Chairperson: -**

**a. Dr. Rakhi Gupta,** Assistant Professor, Dept. of Information Technology, KC College, HSNC University, rakhi.gupta@kccollege.edu.in, 9619914191

2. **Name of Co-chairperson: -**
   a. **Ms. Geeta N. Brijwani,** Assistant Professor, Dept. of Comp. Sci., KC College, HSNC University, geeta.brijwani@kccollege.edu.in, 9890857969

3. **Two to five teachers each having minimum five years teaching experience amongst the full time teachers of the Departments, in the relevant subject.**
   a. **Mrs. Neha Patel,** Assistant Professor, Dept. of Information Technology, KC College, HSNC University, neha.patel@kccollege.edu.in, 9820609142
   b. **Mrs. Nashrah Gowalker,** Assistant Professor, Dept. of Information Technology, KC College, HSNC University, nashrah.gowalker@kccollege.edu.in, 9664774108
   c. **Ms Aisha Shaikh,** Assistant Professor, Dept. of BVoc Web Technologies, KC College, HSNC University, aisha.shaikh@kccollege.edu.in, 9224261350
   d. **Mr. Naveen Pahuja,** Assistant Professor, Dept. of Comp. Sci., KC College, HSNC University, naveenpahuja94@gmail.com, 8856881398
4. **One Professor / Associate Professor from other Universities or professor / Associate Professor from colleges managed by Parent Body; nominated by Parent Body;-**
   a. **Prof. (Dr.) Sushil Kulkarni,** Professor & Dean, School of Mathematics, Mumbai, sushiltry@gmail.com, 9967770658

5. **Four external experts from Industry / Research / eminent scholar in the field relevant to the subject nominated by the Parent Body.**
   a. **Mr. Ravi Gupta,** Director, Frameboxx 2.0, ravi@frameboxx.in, 9820711434
   b. **Mr. Harish Chandar,** Founder/Director, India Tech International Pvt. Ltd., Mumbai, harishchandarb@gmail.com, 9821528022
   c. **Ms. Reshma Desai,** Assistant Professor, Dept. of Comp. Sci., Thakur College of Science and Commerce, Kandivili, reshma09desai@gmail.com, 9820080427
   d. **Mr. Wilson Rao,** Co-ordinator, Dept. of Information Technology & BVOC, Jai Hind College, Autonomous, wilsonrao@gmail.com, 9821354297
   e. **Mr. Maunash A. Jani**, Software Developer, Genius Lynx, Mumbai, maunash08@gmail.com, 9022155698
   f. **Mr. Ajit Vishwakarma,** Corporate Master Trainer, Managing Director, Vinayavish LLP, Mumbai, ajit@vinayavish.com, 9987230297

6. **Top rankers of the Final Year Graduate and Final Year Post Graduate examination of previous year of the concerned subject as invitee members for discussions on framing or revision of syllabus of that subject or group of subjects for one year.**
   a. **Bende Vaishnavi Vijay**, vaishnavibende5@gmail.com, 7738479980

## B.VOC Web Technologies Program Outcomes

| | |
|---|---|
| PO 01 | **Web Technology Mastery**: Master web technologies for designing and managing websites and applications. |
| PO 02 | **Analytical Problem-Solving**: Apply analytical skills to solve web development challenges. |
| PO 03 | **Clear Communication Skills**: Clearly convey technical concepts in writing and speech. |
| PO 04 | **User-Centered Design**: Integrate user experience and design principles for functional web applications. |
| PO 05 | **Industry-Ready Skills**: Gain expertise in AI, Machine Learning, Search Engine Optimization, and Digital Marketing. |
| PO 06 | **Web Security Practices**: Implement security practices and testing for reliable applications. |
| PO 07 | **Ethics and Professionalism**: Demonstrate ethical practices in web development and business. |
| PO 8 | **Teamwork & Leadership**: Develop teamwork and leadership skills for solving web development problems. |
| PO 09 | **Innovative Web Solutions**: Engage in research and create innovative web applications. |
| PO 10 | **Entrepreneurial Mind-set and Employability**: Use entrepreneurial principles to start and manage tech Businesses. Apply skills to enhance career opportunities or start a business. |

**SEMESTER 3**

| | **Subject** | **Lecture/Practical** | **No. of Credits** |
|---|---|---|---|
| Major | Database Management System | 45L/30P | 4(3+1) |
| Major | Software Engineering | 45L/30P | 4(3+1) |
| Minor | Mathematics for Competitive Exams-I | 45L/30P | 4(3+1) |
| SEC | Client-Side Scripting | 30L/30P | 3(2+1) |
| MIL | | 30L | 2 |
| GE | | | 3 |
| OJT/FP/ RP/ CEP/CC | | | 2 |
| | **Total** | | **22** |

**SEMESTER 4**

| | **Subject** | **Lecture/Practical** | **No. of Credits** |
|---|---|---|---|
| Major | Web Programming using Python | 45L/30P | 4(3+1) |
| Major | Software Testing | 45L/30P | 4(3+1) |
| Minor | Mathematics for Competitive Exams-II | 45L/30P | 4(3+1) |
| SEC | Web Programming using PHP | 30L/30P | 3(2+1) |
| MIL | | 30L | 2 |
| GE | | | 3 |
| OJT/FP/ RP/ CEP/CC | | | 2 |
| | **Total** | | **22** |

## The Scheme of Teaching and Examination is as under NEP:

| Year | Sem. | Theory / Practical | Paper Code | Course Title | No of Credits | No of Lectures Hours | Total Credits |
|---|---|---|---|---|---|---|---|
| 1 | I | M1 | VWT101B | Programming with C | 4 | 60 | Major = 8<br>Minor =4 |
| | | M2 | VWT102B | Introduction to Web Designing & Application -I | 4 | 60 | |
| | | Mi3 | VWT103B | Discrete Mathematics | 4 | 60 | |
| | II | M4 | VWT104B | Object Oriented Programming with C++ | 4 | 60 | Major = 8<br>Minor =4 |
| | | M5 | VWT105B | Introduction to Web Designing & Application -II | 4 | 60 | |
| | | Mi6 | VWT106B | Statistics | 4 | 60 | |
| 2 | III | M7 | VWT201B | Database Management System | 4 | 60 | Major = 8<br>Minor =4 |
| | | M8 | VWT202B | Software Engineering | 4 | 60 | |
| | | Mi9 | VWT203B | Mathematics for Competitive Exams-I | 4 | 60 | |
| | IV | M10 | VWT204B | Web Programming using Python | 4 | 60 | Major = 8<br>Minor =4 |
| | | M11 | VWT205B | Software Testing | 4 | 60 | |
| | | Mi12 | VWT206B | Mathematics for Competitive Exams-II | 4 | 60 | |

## Vocational Courses Linked to Major/Minor

| Year | Sem. | Papers | Paper Code | Course Title | No of Credits | No of Lectures Hours | Total Credits |
|---|---|---|---|---|---|---|---|
| 1 | I | V1 | VWT107D | Digital Art and Illustration | 1 | 30 | 1 |
| | II | V2 | VWT108D | Design and Web Layout | 1 | 30 | 1 |

## Indian Knowledge Systems

| Year | Sem. | Papers | Paper Code | Course Title | No of Credits | No of Lectures Hours | Total Credits |
|---|---|---|---|---|---|---|---|
| 1 | I | IKS1 | IKS101D | Introduction To Indian Knowledge Systems - I | 1 | 15 | 1 |
| | II | IKS2 | IKS102D | Introduction To Indian Knowledge Systems - II | 1 | 15 | 1 |

## Value Added Course

| Year | Sem. | Papers | Paper Code | Course Title | No of Credits | No of Lectures Hours | Total Credits |
|---|---|---|---|---|---|---|---|
| 1 | I | VAC1 | GBX101C | Contemporary India: Values and Issues -I | 2 | 30 | 2 |
| | II | VAC2 | GBX103C | Contemporary India: Values and Issues -II | 2 | 30 | 2 |

## Ability Enhancement Course

| Year | Sem. | Papers | Paper Code | Course Title | No of Credits | No of Lectures Hours | Total Credits |
|---|---|---|---|---|---|---|---|
| 1 | I | AEC1 | ENG107C | Communication Skills in English - I | 2 | 30 | 2 |
| | II | AEC2 | ENG108C | Communication Skills in English - II | 2 | 30 | 2 |

# Skill Enhancement Course

| Year | Sem. | Papers | Paper Code | Course Title | No of Credits | No of Lectures Hours | Total Credits |
|---|---|---|---|---|---|---|---|
| 1 | I | SEC I | VWT201C | Client-Side Scripting | 3 | 60 | 3 |
| | II | SEC II | VWT202C | Web Programming using PHP | 3 | 60 | 3 |

# Detailed Scheme Theory Semester III

## MAJOR SUBJECT - 1

## Database Systems

**Course Objectives**

- To introduce the fundamental concepts of Database Management Systems (DBMS) and their advantages over traditional file systems.
- To familiarize students with various data models, including relational, hierarchical, and network models, and their applications.
- To provide a strong understanding of the Entity-Relationship (ER) model for conceptual database design.
- To introduce relational database concepts, including normalization techniques for designing efficient databases.
- To develop skills in Structured Query Language (SQL) for data definition, manipulation, and access control.

**Course Outcomes**

- To introduce the fundamental concepts, architecture, and significance of Database Management Systems (DBMS).
- To explain different data models, including relational, hierarchical, and network models, and their practical applications.
- To develop an understanding of the Entity-Relationship (ER) model for designing structured and efficient databases.
- To explore the relational data model, including domains, attributes, tuples, relations, and integrity constraints.
- To familiarize students with normalization techniques for database optimization and redundancy reduction.
- To equip students with SQL skills for database creation, manipulation, querying, and access control.
- Implementation of tables, functions and sub-queries.
- Implementation of views and DCL statements.

| Unit | Content | No. of Lectures |
|---|---|---|
| 1 | **1.1 Introduction to DBMS:** Database, DBMS – Definition, Overview of DBMS, Advantages of DBMS, Levels of abstraction, Data independence, DBMS Architecture<br>**1.2 Data models:** Client/Server Architecture, Object Based Logical Model, Record Based Logical Model ( relational, hierarchical, network)<br>**1.3 Entity Relationship Model:** Entities, attributes, entity sets, relations, relationship sets, Additional constraints ( key constraints, participation constraints, weak entities, aggregation / generalization, Conceptual Design using ER ( entities VS attributes, Entity Vs relationship, binary Vs ternary, constraints beyond ER) | 15 |

| | | |
|---|---|---|
| **2** | **2.1 Relational data model:** Domains, attributes, Tuples and Relations, Relational Model Notation, Characteristics of Relations, Relational Constraints - primary key, referential integrity, unique constraint, Null constraint, Check constraint<br>**2.2 Normalization Concepts:** 1NF, 2NF, 3NF, BCNF, examples.<br>**2.3 DDL Statements**: Creating Databases, Using Databases, data types, Creating Tables (with integrity constraints – primary key, default, check, not null), Altering Tables, Renaming Tables, Dropping Tables, Truncating Tables, Backing Up and Restoring databases<br>**2.4 DML Statements:** Viewing the structure of a table insert, update, delete, Select all columns, specific columns, unique records, conditional select, in clause, between clause, limit, aggregate functions (count, min, max, avg, sum), group by clause, having clause<br>**2.5 DCL Statements:** (creating/dropping users, privileges introduction, granting/revoking privileges, viewing privileges) | **15** |
| **3** | **3.1 Functions:** String Functions (concat, instr, left, right, mid, length, lcase/lower, ucase/upper, replace, strcmp, trim, ltrim, rtrim), Math Functions (abs, ceil, floor, mod, pow, sqrt, round, truncate) Date Functions (adddate, datediff, day, month, year, hour, min, sec, now, reverse)**3.1 Joining Tables:** inner join, outer join (left outer, right outer, full outer)<br>**3.2 Sub queries:** Sub queries with IN, EXISTS, sub queries restrictions, Nested sub queries, ANY/ALL clause, correlated subqueries<br>**3.3 Views:** Creating, altering dropping, renaming and manipulating views | **15** |

**SLE Topics**

| |
|---|
| Comparison of File System vs DBMS |
| Levels of Abstraction in DBMS |
| Types of Database Models |
| ER Model in Depth |
| SQL DDL vs DML vs DCL |
| Joins in SQL |
| Views in SQL |
| Real-World Applications of DBMS |

**Online Resources**

| |
|---|
| **https://nptel.ac.in/courses/106105175**<br>**https://archive.nptel.ac.in/courses/106/105/106105175/** |

**Reference Books**


1. Ramez Elmasri & Shamkant B.Navathe, Fundamentals of Database Systems, Pearson Education, Sixth Edition, 2010
2. Ramakrishnam, Gehrke, Database Management Systems, McGraw Hill, 2007
3. Joel Murach, Murach's MySQL, Murach, 2012.
4. Robert Sheldon, Geoff Moes, Beginning MySQL, Wrox Press, 2005

**Practicals: Database Systems**

| Practical No. | Practical |
|---|---|
| **1.** | For a given scenario, Draw E-R diagram and convert entities and relationships to a table. |
| **2.** | Design a database for any case study with tables and normalize using normalization concept. |
| **3.** | Perform the following: Viewing all databases, Creating a Database, Viewing all Tables in a Database |
| **4.** | Creating Tables (With and Without Constraints), Inserting/Updating/Deleting Records in a Table, Saving (Commit) and Undoing (rollback) |
| **5.** | Perform the following: Altering a Table, Dropping/Truncating/Renaming Tables, Backing up / restoring a Database. |
| **6.** | Perform the following: Simple Queries, Simple Queries with Aggregate functions, Queries with Aggregate functions (group by and having clause) |
| **7.** | Queries involving Date Functions, String Functions, Math Functions, Join Queries, Inner Join, Outer Join |
| **8.** | Sub-queries: With IN clause, With EXISTS clause |
| **9.** | Views :Creating Views (with and without check option), Dropping views, Selecting from a view |
| **10.** | DCL statements : Granting and revoking permissions |

# MAJOR SUBJECT – 2

## Software Engineering

**Course Objectives**

- The objective of this course is to provide students with an understanding of Software engineering principles, processes, and methodologies.
- To orient them with the idea of decomposing the given problem into Analysis, Design, Implementation, Testing and Maintenance phases.
- To provide an idea of using various process models in the software industry according to given problem statement and circumstances.
- To gain the knowledge of how Analysis, Design, Implementation, Testing and Maintenance processes are conducted in a software project.

**Course Outcomes**

- Understand the principles and practices of software engineering.
- Students will be able to decompose the given project in various phases of a lifecycle.
- Apply software engineering processes and methodologies to develop software systems.
- Demonstrating competence in communication, planning, analysis, design, construction and Deployment.
- Implement software using appropriate programming languages and development tools.

- Students will be able to choose appropriate process model depending on the user requirements.
- Analyse and able to work as individual and as a part of multidisciplinary team to develop and deliver quality software.
- Students can apply the knowledge, techniques, and skills in the development of a software product.

| Unit | Content | No. of Lectures |
|---|---|---|
| 1 | **1.1 Introduction:** What is software engineering? Software Development Life Cycle, Requirements Analysis, Software Design, Coding, Testing, Maintenance etc.<br>**1.2 Software Requirements:** Functional and Non-functional requirements, User Requirements, System Requirements, Interface Specification, Documentation of the software requirements.<br>**1.3 Software Processes:** Process and Project<br>**1.4 Software Development Process Models.**<br>• Waterfall Model.<br>• Prototyping.<br>• Iterative Development /Spiral Model  Rational Unified Process.<br>• The RAD Model<br>• Time boxing Model.<br>**1.5 Agile software development**: Agile methods, Plan-driven and agile development, Extreme programming, Agile project management.<br>**1.6 Socio-technical system:** Essential characteristics of socio technical systems, Emergent System Properties, Systems Engineering, Components of system such as organization, people and computer, Legacy Systems.<br>**1.7 Critical system:** Types of critical system, A simple safety critical system, Dependability of a system, Availability and Reliability, Safety and Security of Software systems. | **15** |
| 2 | **2.1 Requirements Engineering Processes:** Feasibility study, Requirements elicitation and analysis, Requirements Validations, Requirements Management<br>**2.2 Architectural Design:** Architectural Design Decisions, System Organization, Modular Decomposition Styles, Control Styles, Reference Architectures.<br>**2.3 User Interface Design:** Need of UI design, Design issues, The UI design Process, User analysis, User Interface Prototyping, Interface Evaluation.<br>**2.4 Project Management**: Software Project Management, Management activities, Project Planning, Project Scheduling, Risk Management<br>**2.5**. **Quality Management:** Process and Product Quality, Quality assurance and Standards, Quality Planning, Quality Control, Software Measurement and Metrics. | **15** |

| | | |
|---|---|---|
| | **2.6 Verification and Validation:** Planning Verification and Validation, Software Inspections, Automated Static Analysis, Verification and Formal Methods. V-model for software, Testing during stages of V-model, VV Model, Critical Roles and Responsibilities. Acceptance Testing. | |
| **3** | **3.1 Fundamentals & Principles of testing**: Important Features of Testing Process, Unit Testing, Component Testing System Testing, Test Case Design, Test Automation.<br>**3.2 Testing Strategy:** Categories of Defect, Defect, Error, or Mistake in Software, Developing Test Strategy, Developing Testing Methodologies (Test Plan)/ Approaches, Testing Process,<br>**3.3 Software Maintenance:** Management of maintenance, Maintenance process, Maintenance models, Regression testing, Reverse engineering, Software reengineering, Configuration management, documentation.<br>**3.4 Process Improvement:** Process and product quality, Process Classification, Process Measurement.<br>**3.5 Service Oriented Software Engineering:** Services as reusable components, Service Engineering, Software Development with Services.<br>**3.6 Software reuse:** The reuse landscape, Application frameworks, Software product lines, COTS product reuse. | **15** |

**SLE Topics**

| |
|---|
| Introduction to Software Engineering and the Software Development Life Cycle (SDLC) |
| Functional vs Non-Functional Requirements and Their Documentation |
| Software Development Process Models: Waterfall, Spiral, RAD, and More |
| Agile Software Development and Extreme Programming |
| Socio-Technical Systems and Critical System Dependability |
| Requirements Engineering Processes and Architectural Design |
| User Interface Design Principles, Prototyping, and Evaluation |
| Software Project and Quality Management Practices |
| Software Testing Fundamentals, Strategies, and the V-Model |
| Software Maintenance, Reuse, and Process Improvement Techniques |

**Online Resources**

| |
|---|
| https://www.coursera.org/courses?query=software%20engineering |
| https://www.edx.org/micromasters/software-development |
| https://www.geeksforgeeks.org/software-engineering/ |
| https://www.tutorialspoint.com/software_engineering/index.htm |
| https://ieeexplore.ieee.org/Xplore/home.jsp |
| https://www.scrum.org/resources/what-is-scrum |

**Reference Books:**


1. Software Engineering by Ian Somerville, Pearson Education. Ninth Edition.
2. Software Engineering by Pankaj Jalote, Narosa Publication
3. Software engineering, a practitioner's Approach by Roger Pressman, Tata Mcgraw-Hill, Seventh Edition.
4. Software Engineering principles and practice by WS Jawadekar, Tata Mcgraw-hill

5. Software Testing and Continuous Quality Improvement by William E. Lewis, CRC Press, Third Edition, 2016
6. Software Testing: Principles, Techniques and Tools by M. G. Limaye, TMH, 2017.
7. Foundations of Software Testing by Dorothy Graham, Erik van Veenendaal, Isabel Evans, Rex Black, Cengage Learning, Third Edition.
8. Software Testing Technique by Boris Beizer, Dreamtech Press, Premier press 2014

9. Software Testing: A Craftsman"s Approach by Paul C. Jorgenson, CRC Press, Fourth Edition 2017 Software Testing A Craftsman's approach by Paul C. Jorgensen, CRC Press, Second Edition 1997


**Practical List**

| Practical no. | Practical |
|---|---|
| 1. | Study and implementation of Entity Relationship Diagrams |
| 2. | Study and implementation of Data Flow Diagrams. |
| 3. | Study and implementation of Class diagrams |
| 4. | Study and implementation of Use Case Diagrams |
| 5. | Study and implementation of Sequence Diagrams |
| 6. | Study and implementation of State Transition Diagrams. |
| 7. | Study and implementation of. Activity Diagrams |
| 8. | Effort & Cost estimation (case study) |

# MINOR SUBJECT

## Mathematics for Competitive Exams-I

**Course Objectives**

- Develop a strong foundation in Numerical Ability, covering topics like numbers, algebra, percentages, and probability.
- Enhance Reasoning Ability through logical puzzles, Venn diagrams, seating arrangements, and ranking problems.
- Build Data Interpretation skills to analyze bar graphs, pie charts, and tabular data.
- Improve problem-solving and critical thinking for real-world applications and competitive exams.
- Strengthen mathematical and logical reasoning techniques for academic and professional success.

**Course Outcomes**

- Ability to solve numerical problems related to HCF, LCM, percentages, and algebra.
- Improved logical reasoning skills to tackle classification, syllogisms, and seating arrangement problems.
- Enhanced data analysis skills for interpreting information from various graphical representations.
- Assess the sufficiency of given data and apply critical thinking to make informed decisions in data interpretation and reasoning-based scenarios.
- Strengthened analytical and problem-solving abilities for exams and career growth.
- Interpret, analyze, and draw conclusions from various data representations including bar graphs, pie charts, line charts, and tabular data.
- Practical application of mathematics and reasoning in real-life scenarios.
- Implementation of data interpretation and data analysis.

| Unit | Content | No of Lecture hours |
|---|---|---|
| **1** | **Numerical Ability**: Numbers, HCF & LCM, Average, Mathematical Operator, Problems on Ages, Percentage, Profit and Loss, Algebraic Expressions and Inequalities, Trigonometry, Probability | **15** |
| **2** | **Reasoning Ability**: Classification, Logical Venn diagram, Cube and Dice, Advance Puzzles, Seating Arrangements, Distance and Direction, Blood Relations, Syllogism, Order and Ranking, Arithmetic Reasoning. | **15** |
| **3** | **Data Interpretation and Data Analysis and Sufficiency**<br>Bar Graph<br>Line Chart<br>Tabular Data<br>Pie Chart | **15** |

**SLE Topics**

| |
|---|
| Age-Based Puzzles with Data Sufficiency<br>Profit-Loss with Tabular Data Interpretation<br>Trigonometry-Based Distance & Direction Problems<br>Venn Diagram with Probability and Classification<br>Seating Arrangements with Inequalities and Ranking |

**Online Resources**

| |
|---|
| **https://d5ofvi41ggben.cloudfront.net/4966d784-a71b-4c11-b31e-801ee59d95c0-1571828515642-quantitative-aptitude.pdf** |

**References**

1. Quantitative Aptitude for Competitive Examinations - Quantitative Aptitude R.S Agrawal, S.Chand,ISBN: 9789352534029, 9789352534029Edition: Revised & Enlarged Edition, 2020

2. Verbal Reasoning (Useful For Various Competitive Exams), Dr. LAL & KUMAR, ISBN: 978-81- 7482-581-0

**Practical List**

| Practical | Content |
|---|---|
| 1 | Practical on Inequalities and Quadratic Equation |
| 2 | Practical on Number Series |
| 3 | Practical to calculate Profit and Loss |
| 4 | Practical to calculate Interest |
| 5 | Practical to calculate Speed, Distance and Time, Work, Time and Wages. |
| 6 | Practical on Data Interpretation<br>6a. Bar Graph<br>6b. Line Chart |
| 7 | Practical on Data Interpretation<br>7a. Tabular Data<br>7b. Pie Chart |
| 8 | Practical on Venn diagrams |

## Detailed Scheme Theory Semester III

## MAJOR SUBJECT - 1

## Web Programming using Python

**Course Objectives**

- To introduce students to Python programming, its history, features, and execution modes.
- To develop an understanding of data types, expressions, operators, and built-in functions.
- To familiarize students with control structures, functions, and modular programming concepts.
- To enable students to perform file handling, exception handling, and GUI programming using Tkinter.
- To provide hands-on experience with database connectivity using MySQL in Python.
- To develop problem-solving skills using Python programming constructs.

**Course Outcomes**

Upon completion of the course, students will be able to:

- Explain Python's core concepts, including data types, variables, and operators.

- Implement control structures such as loops and conditional statements to develop logical programs.
- Write and utilize functions and modules to promote code reusability and efficiency.
- Perform file handling operations and implement exception handling mechanisms.
- Develop GUI applications using Tkinter and understand various widgets and event handling.
- Integrate Python with databases using MySQL and execute SQL queries programmatically.
- Implementation of statements, functions and dictionaries.
- Implement the concept of inheritance, exception handling and regular expressions

| Unit | Content | No. of Lecture hours |
|---|---|---|
| 1 | **Introduction**: The Python Programming Language, History, features, Installing Python, Running Python program. Interactive and script modes of IDLE, Data Types: Values and Types Type conversion, expressions and operators of type int, float, boolean. Built-in function type. Operator precedence. Variables, Variable Names and Keywords.<br>**Strings**: Strings and tuples are immutable, lists are mutable. String Methods, operators and comparison.<br>**Tuples**: Built-in methods, Operations<br>**Lists:** Accessing elements, Built-in List functions, List Operations<br>**Sets and Dictionaries**: Difference between sets and dictionaries, Sets and frozen sets. Creating a Dictionary, Accessing Values in a Dictionary, Built-in methods, Operations on dictionary. | **15** |
| 2 | **Control Statements**: The conditional statements if, if-else, if-elsif-else The iterative statements while, while-else, for-else. Nested compound statements. The continue statement to skip over one iteration of a loop, the break statement to exit the loop, pass statement.<br>**Functions**: The import statement for already-defined functions and constants. The compound statement def to define functions; the role of indentation for delimiting the body of a compound statement; calling a previously defined function. Advantages of functions, function parameters, Recursive functions Built-in functions.<br>**Modules**: Importing module, Creating and exploring modules, Math module, Random module, Time module. Use of modules and Packages, namespaces, scopes, objects and classes | **15** |
| 3 | **Python File Input-Output**: Opening and closing files, Various types of file modes, Reading and writing to files.<br>**Exception handling**: What is an exception, Various keywords to handle exceptions such try, catch, except, else, finally, raise.<br>**GUI with Tkinter**: Widgets- Button, Canvas, Checkbutton, Entry, Frame, Label, Listbox, Menubutton, Menu, Radiobutton, Scale, Scrollbar, Text. Spinbox, PanedWindow, LabelFrame, tkMessagebox. Handling Standard attributes and Properties of Widgets.<br>**Introduction to Flask:** Installation of Flask and setting up the environment, Understanding routing and HTTP method, using CSS and Bootstrap with Flask, Handling form submission and validation, | **15** |

| | | |
|---|---|---|
| | Introduction to Database using SQL Lite/MySQL, Connecting Flask to database | |

**SLE Topics**

| |
|---|
| Flask session and cookies<br>Error handling and custom error page |

**Online Resources**

| |
|---|
| **https://nptel.ac.in/courses/106106145** |

**Reference Books:**


1. Get hands-on with Python Programming and Django web development ,Fabrizio Romano , Gastón C. Hillar , Arun Ravindran


**Practical List**

| Practical No. | Practical |
|---|---|
| 1. | Programs based on lists, conditional constructs, the for statement and the range function; interactively using the built-in functions len, sum, max, min. |
| 2 | 2a. Programs using break and continue statements.<br>2b. Write a program to implement user defined module |
| 3 | 3a. Programs related to string manipulation.<br>3b. Programs using list comprehensions and anonymous functions |
| 4 | 4a. Programs related to dictionaries.<br>4b. Programs using the built-in methods of the string, list and dictionary classes |
| 5 | Design a class that store the information of Employee and display the same. |
| 6 | Implement the concept of inheritance using python. |
| 7 | 7a. Programs to read and write files.<br>7b. Program to demonstrate exception handling |
| 8 | Program to demonstrate the use of regular expressions |
| 9 | Program based on GUI application using Tkinter. |

# MAJOR SUBJECT – 2

## Software Testing

**Course Objectives**

- To introduce students to the fundamentals of software testing, including goals, methodologies, and life cycle models, ensuring a solid understanding of verification and validation processes.
- To develop proficiency in dynamic and static testing techniques, such as black-box, white-box, regression, and mutation testing, along with effective test management and planning.

- To equip students with knowledge of software metrics and efficient test suite management, focusing on monitoring, controlling, and improving the testing process.
- To familiarize students with automation tools and continuous integration platforms like Selenium and Jenkins, enabling them to apply automation in real-world testing scenarios.

**Course Outcomes**

- Understand and explain the principles, goals, and methodologies of software testing and apply them in various software development environments.
- Demonstrate the ability to design and execute test plans using the Software Testing Life Cycle (STLC) and ensure thorough verification and validation.
- Apply dynamic testing techniques such as boundary value analysis, equivalence partitioning, and state-based testing to identify defects effectively.
- Implement white-box testing methods, including logic coverage, basis path testing, and data flow testing to ensure comprehensive code validation.
- Manage and optimize test suites through minimization, prioritization, and efficient resource allocation, ensuring cost-effective testing processes.
- Utilize software metrics to monitor and control the testing process, and apply estimation models to predict testing effort and quality outcomes.
- Employ automation tools like Selenium, QTP, and LoadRunner to automate test cases, improve efficiency, and reduce manual errors in testing.
- Integrate continuous testing into delivery pipelines using Jenkins, understanding how to manage Jenkins environments and integrate with automation tools for seamless deployment.

| Unit | Content | No. of Lectures |
|---|---|---|
| 1 | **1.1 Testing Introduction :** Goals of Software Testing, Software Testing Methodology Definitions, Model for Software Testing, Effective Software Testing vs Exhaustive Software Testing, Software Failure Case Studies, Software Testing Terminology,<br>**1.2 Software Testing Life Cycle (STLC),** Software Testing methodology, Verification and Validation, Verification requirements, Verification of high-level design, Verification of low-level design, validation.<br>**1.3 Test Management:** test organization, structure and of testing Managing the group, test planning, detailed test design and test, Test Process specification. | **15** |
| 2 | **2.1 Dynamic Testing:** Black Box testing: boundary value Techniques analysis, equivalence class testing, state table based testing, cause-effect graphing based testing, error guessing.<br>**2.2 White box Testing Techniques:** need, logic coverage criteria, basis path testing, graph matrices, loop testing, data flow testing, mutation testing. Static Testing.<br>**2.3 Validation Activities**: Unit validation, Integration, Function, System, Acceptance Testing.<br>**2.4 Regression Testing:** Progressive vs. Regressive, regression testing produces quality software, regression testability, objectives of regression testing, regression testing types, define problem, regression testing techniques. | **15** |

| | | |
|---|---|---|
| **3** | **3.1 Software Metrics**: Need, definition and classification of software matrices. Testing Metrics for Monitoring and Controlling the Testing Process: attributes and corresponding metrics, estimation model for testing effort, architectural design, information flow matrix used for testing, function point and test point analysis.<br>**3.2 Efficient Test Suite Management:** minimizing the test suite and its benefits, test suite minimization problem, test suite prioritization its type, techniques and measuring effectiveness<br>**3.3 Test Automation and Testing Tools:** need, categorization, Automation selection and cost in testing tool, guidelines for testing tools. Study of testing tools: WinRunner, QTP, LoadRunner, Test Director and IBM Rational Functional Tester, Selenium etc.<br>**3.4 Jenkins: Introduction** to delivery pipeline, Introduction to Jenkins, Jenkins management, adding slave node to Jenkins, Building a delivery pipeline, Selenium integration with Jenkins | **15** |

**SLE Topics**

| |
|---|
| Introduction to Software Testing: Goals, Methodologies, and Terminologies<br>The Software Testing Life Cycle (STLC) and Verification & Validation Techniques<br>Dynamic Testing Techniques: Black Box Testing Methods and Applications<br>White Box Testing Techniques: Logic Coverage, Basis Path, and Mutation Testing<br>Regression Testing: Techniques, Types, and Best Practices for Quality Software<br>Software Metrics: Monitoring, Controlling, and Estimating Testing Efforts<br>Test Automation and Tools: Overview of Selenium, QTP, LoadRunner, and More<br>Continuous Integration with Jenkins: Building Pipelines and Selenium Integration |

**Online Resources**

| | |
|---|---|
| 1 | https://www.guru99.com/software-testing.html |
| 2 | https://www.softwaretestinghelp.com/ |
| 3 | https://www.coursera.org/courses?query=software%20testing |
| 4 | https://www.edx.org/learn/software-testing |
| 5 | https://selenium.dev/documentation/ |
| 6 | https://www.jenkins.io/doc/ |

**References**


1. Software Testing Principles and Practices Naresh Chauhan Oxford Higher Education
2. Effective Methods for Software Testing , third edition by Willam E. Perry, Wiley Publication
3. Software Testing and quality assurance theory and practice by Kshirasagar Naik, Priyadarshi Tripathy , Wiley Publication
4. Software Testing Concepts and Tools by Nageswara Rao Pusuluri , dreamtech press


**Practical list**

| Practical No. | Practical |
|---|---|
| **1.** | Evaluating Test Exit Criteria and Reporting |

| | |
|---|---|
| **2.** | Static and Dynamic Analysis |
| **3.** | Rate Quality Attributes for Domain and Technical Testing |
| **4.** | Draw State Graph |
| **5.** | Incident Management |
| **6.** | Path Testing and Equivalence Partitioning |
| **7.** | Performance Testing |
| **8.** | Using Testing Tool Selenium |
| **9.** | Using Testing Tool QTP |
| **10.** | Using Testing Tool WAPT |

# MINOR SUBJECT

## Mathematics for Competitive Exams-II

**Course Objectives**

- Develop a strong foundation in Numerical Ability, covering ratio & proportion, time & work, speed & distance, and financial calculations.
- Enhance Reasoning Ability through coding-decoding, number series, analogy, and decision-making techniques.
- Build Research Aptitude by understanding research methods, thesis writing, and referencing styles.
- Improve problem-solving and analytical skills for real-world numerical and logical applications.
- Prepare students for competitive exams and research-oriented careers by strengthening quantitative, logical, and analytical abilities.

**Course Outcomes**

- Ability to solve numerical problems related to ratios, time-work equations, speed-distance calculations, and financial mathematics.
- Analyze patterns and sequences using coding-decoding, alphabet-number series, analogies, and machine input-output problems.
- Improved logical reasoning skills for analyzing coded information, number sequences, and decision-making scenarios.
- Enhanced research skills, including understanding research methods, writing formats, and ethical considerations.
- Strengthened critical thinking and problem-solving abilities for academic and career growth.
- Apply knowledge of research processes, thesis/article writing, ICT tools, and research ethics to academic and professional research activities.
- Practical application of mathematical and reasoning concepts in real-life and professional settings.

- Implementation of research methods, thesis and application of ICT in research

| Unit | Content | No of Lecture hours |
|---|---|---|
| **1** | **Numerical Ability**: Ratio and Proportion Partnership Time and work Pipes and cisterns Time and distance Problems on trains Boats and streams Simple and Compound interest | **15** |
| **2** | **Reasoning Ability**: Coding-Decoding, Machine Input-Output, Alphabet and Number Series, Analogy, Passage and Conclusions, Statement and Conclusion, Statement and Assumptions, Statement and Arguments, Decision Making, Coded Inequalities. | **15** |
| **3** | **Research Aptitude:** Research: Meaning, Types, and Characteristics, Positivism and Postpositivism approach to research, Methods of Research: Experimental, Descriptive, Historical, Qualitative and Quantitative methods, Steps of Research, Thesis and Article writing: Format and styles of referencing, Application of ICT in research, Research ethics. | **15** |

**SLE Topics**

| |
|---|
| Compound Problem Solving using Ratio, Proportion & Time Concepts<br>High-Level Input-Output & Coding-Decoding Systems<br>Quantitative Data Interpretation with Interest and Speed Problems<br>Critical Thinking through Logical Statements & Arguments<br>Integrated Research Aptitude Analysis with ICT Application |

**Online Resources**

| |
|---|
| **https://d5ofvi41ggben.cloudfront.net/4966d784-a71b-4c11-b31e-801ee59d95c0-1571828515642-quantitative-aptitude.pdf** |

**Practical List**

| Practical No. | Practical |
|---|---|
| 1 | Practical on Time, Speed, and Distance |
| 2 | Practical on Simple and Compound Interest |
| 3 | Practical on Coding-Decoding |
| 4 | Practical on Alphabet and Number Series |
| 5 | Practical on Statement and Conclusion |
| 6 | **Practical on Research Methods** – Identify and classify research methods as experimental, descriptive, historical, qualitative, or quantitative. |
| 7 | **Practical on Thesis and Article Writing** – Draft a small research report using proper format and referencing styles. |
| 8 | **Practical on Application of ICT in Research** – Use MS Excel, Google Scholar, or other research tools for data analysis and referencing. |

**References**

1. Quantitative Aptitude for Competitive Examinations - Quantitative Aptitude R.S Agrawal, S.Chand,ISBN: 9789352534029, 9789352534029Edition: Revised & Enlarged Edition, 2020

2. Verbal Reasoning (Useful For Various Competitive Exams), Dr. LAL & KUMAR, ISBN: 978-81- 7482-581-0

## Skill Enhancement Course

## Semester-III

# Client-Side Scripting

**Course Objectives**

- To provide a foundational understanding of JavaScript, including its core concepts, event handling, and asynchronous programming.
- To develop proficiency in DOM manipulation and client-side scripting for dynamic web applications.
- To introduce AngularJS and its MVC architecture, covering data binding, directives, and routing for building structured web applications.
- To familiarize students with ReactJS, focusing on component-based development, state management, and handling API interactions.
- To enable students to build interactive and responsive web applications using modern frontend technologies.

**Course Outcomes**

- Implement core JavaScript fundamentals and utilize event-driven programming for interactive web development.
- Dynamically manipulate the DOM and manage user interactions using modern JavaScript (ES6+) features.
- Develop single-page applications (SPAs) with AngularJS, incorporating services, directives, and form validation.
- Build ReactJS applications using functional components, hooks, and API integration for dynamic content.
- Apply AngularJS and ReactJS frameworks to create scalable and responsive frontend architectures.
- Leverage modern JavaScript tools and frameworks to deliver robust and user-friendly web applications.

| Unit | Content | No. of Lectures |
|---|---|---|
| 1 | **Unit 1 JavaScript Fundamentals**<br>• Introduction to JavaScript<br>• Variables, Data Types, and Operators<br>• Control Structures (Loops & Conditionals)<br>• Functions and Scope<br>• Objects and Arrays<br>• JavaScript ES6 Features (let, const, arrow functions, spread/rest, destructuring)<br>• Error Handling (try-catch)<br>• Event Handling in JavaScript<br>**Advanced JavaScript & DOM Manipulation**<br>• Document Object Model (DOM)<br>• Selecting and Manipulating Elements<br>• Event Listeners and Delegation<br>• Form Handling and Validation | **15** |
| 2 | **Unit 2 Introduction to AngularJS**<br>• Overview of AngularJS and MVC Architecture<br>• Directives and Data Binding<br>• Controllers and Scope<br>• HTTP Requests in AngularJS<br>• Form Handling and Validation<br>**Introduction to ReactJS**<br>• Overview of ReactJS and Virtual DOM<br>• JSX and Components (Functional & Class)<br>• Props and State Management<br>• Event Handling in React | **15** |

**SLE Topics**

 **JavaScript Advanced Concepts:**

- Implement ES6 features like template literals, destructuring, and arrow functions.
- Write a JavaScript program to handle promises and async-await for API calls.

 **DOM and Event Handling:**

- Develop a real-time search filter that updates the displayed list as the user types.
- Create an interactive image gallery with JavaScript event listeners.

 **AngularJS Directives & Routing:**

- Implement a dynamic form in AngularJS that validates user input and displays errors dynamically.

- Set up a multi-view AngularJS application with routing and lazy loading.

 ReactJS **Component Lifecycle & Hooks:**

- Develop a weather app using React, fetching live weather data via API.
- Create a simple task tracker using React hooks (useState, useEffect).

**Online Resources**

https://onlinecourses.swayam2.ac.in/nou24_cs09/preview

**References:**


1. "Head First JavaScript Programming" by Eric Freeman and Elisabeth Robson
2. "AngularJS: Angular JS in 8 Hours" by Ray Yao
3. "Learning React" by Alex Banks and Eve Porcello
4. "The Road to React" by Robin Wieruch


**Practical List:**

| Practical No. | Practical |
|---|---|
| **1.** | **JavaScript Basics:** Write a JavaScript program to create a calculator that performs basic arithmetic operations (Addition, Subtraction, Multiplication, and Division). |
| **2.** | **Event Handling in JavaScript:** Create a webpage with a button that changes the background color of the page when clicked. |
| **3.** | **DOM Manipulation:** Develop a to-do list where users can add and remove tasks dynamically. |
| **4.** | **Form Validation:** Create a registration form that validates user inputs like email, password length, and phone number format. |
| **5.** | **Asynchronous JavaScript:** Fetch and display data from an API using Fetch API and display it on a webpage. |
| **6.** | **Basic Angular App with Data Binding** |
| **7.** | **Angular App with services and List Rendering** |
| **8.** | **React Components:** Develop a ReactJS application that displays a list of users using functional components and props. |
| **9.** | **React Hooks:** Build a simple counter application using React Hooks (useState). |
| **10.** | **React API Calls:** Create a React application that fetches and displays data from an external API using axios. |

# Semester IV

## Web Programming using PHP

**Course Objectives**

- Learn installation, syntax, variables, data types, and comments.
- Work with arrays, strings, operators, and precedence.
- Use conditionals and loops for flow control.
- Utilize built-in and custom functions, arguments, and string manipulation.
- Process forms, validate input, manage cookies, sessions, and handle errors.

**Course Outcomes**

- Install and configure PHP and XAMPP, and write basic PHP scripts embedded in HTML
- Create and manipulate arrays and strings while understanding operator precedence.
- Use conditional statements and loops to control program execution.
- Develop and utilize built-in and user-defined functions with different argument passing methods.
- Create and process forms, capturing user input and passing data between pages.
- Implement input validation techniques and error handling for robust applications.
- Apply session and cookie handling to maintain state across web pages.

| Unit | Content | No. of Lectures |
|---|---|---|
| 1 | **Unit 1: Introduction to PHP and Web Development**<br>• Overview of Web Programming<br>• Introduction to PHP<br>• Installing PHP, Apache, and MySQL (XAMPP/WAMP)<br>• PHP Syntax and Variables<br>• Data Types, Constants, and Operators<br>• Control Structures (if-else, loops, switch-case)<br>• Functions in PHP<br>• Form Handling and User Input Processing | 15 |
| 2 | **Unit 2: Advanced PHP and Database Integration**<br>• PHP Arrays and String Functions<br>• File Handling in PHP (Read/Write Operations)<br>• Session Management (Sessions and Cookies)<br>• Introduction to MySQL and Database Connectivity<br>• CRUD Operations (Create, Read, Update, Delete) using PHP and MySQL<br>• Error Handling and Exception Handling<br>• Introduction to Object-Oriented PHP | 15 |

## SLE Topics

 **PHP and MySQL Integration:**
Learn how to interact with MySQL databases using PDO and MySQLi.
Implement advanced queries and stored procedures.

 **Object-Oriented Programming in PHP:**
Understand OOP concepts such as classes, objects, inheritance, and polymorphism in PHP.
Implement a small OOP-based application like an e-commerce cart system.

 **PHP Frameworks (Laravel, CodeIgniter):**
Explore PHP frameworks like Laravel or CodeIgniter for building scalable applications.
Develop a basic Laravel project with authentication and CRUD operations.

 **RESTful API Development with PHP:**
Learn how to create and consume REST APIs using PHP.
Develop an API to fetch and update data from a MySQL database.

 **Security Best Practices in PHP:**
Implement CSRF protection, input validation, and data sanitization techniques.
Learn how to securely store passwords using hashing techniques like bcrypt.

## Online Resources

PHP.net - https://www.php.net/manual/en/index.php

W3Schools - https://www.w3schools.com/php/

Codecademy - https://www.codecademy.com/learn/learn-php

PHP: The Right Way - https://www.phptherightway.com/

Udemy - https://www.udemy.com/courses/search/?q=php

**References:**


1. HTML 5 Black Book, Covers CSS 3, JavaScript, XML, XHTML, AJAX, PHP and jQuery, 2ed, Dreamtech Press
2. Web Programming and Interactive Technologies, scriptDemics, StarEdu Solutions India. PHP: A Beginners Guide, Vikram Vaswani, TMH
3. Learning PHP, MySQL, JavaScript, CSS & HTML5, Robin Nixon, O'Reilly.
4. PHP, MySQL, JavaScript & HTML5 All-in-one for Dummies, Steve Suehring, Janet Valade Wiley

## Practical List

| Practical No. | Practical |
|---|---|
| 1. | a. Write a PHP script, to check whether the page is called from 'https' or 'http'.<br>b. Write a PHP script to redirect a user to a different page. |
| 2. | a. Write a PHP script, which changes the colour of the first character of a word.<br>b. Write a PHP script to count the number of lines in a file. |
| 3. | Write a PHP program to print out the multiplication table up to 6*6. |
| 4. | Write a PHP program to remove duplicates from a sorted list. |
| 5. | Write a simple PHP program to check that emails are valid. |
| 6. | Write a simple PHP program to verify that the user did not leave any fields blank when submitting the form. |
| 7. | Check to make sure the credit card number is composed of exactly 16 numerical digits. Check to make sure that a Visa card number starts with a "4" and a MasterCard number starts with a "5". |
| 8. | Write a program to demonstrate regular expression |
| 9. | Working with Sessions and Cookies:<br>a (Storing Records / Reprieving Records and Display them)<br>b. Storing and Retrieving Cookies<br>c. Storing and Retrieving Sessions |
| 10. | Database CRUD Operations: Create a simple student management system where users can add, update, delete, and view student records using PHP and MySQL. |