



Shiv Chhatrapati Shikshan Sanstha's

**Rajarshi Shahu Mahavidyalaya, Latur**  
**(Autonomous)**

**Curriculum**

**2022-2023**

**B.Sc. C.S.**

(CC/AECC/SEC/GE)

**UG First Year Semester I & II**

**Under CBCS**

**Three Year Degree Programme in B.Sc.C.S.**

**(Six Semester Course)**

Syllabi approved by the Board of Studies in

B.Sc.C.S. with effect from June, 2022

**Rajarshi Shahu Mahavidyalaya(Autonomous), Latur**  
**UG I**  
**Semester Pattern**  
**Semester I & II**

**Semester: I**

Course Code	Course Title	Lectures		Marks		Credits
		Per Week	Total	Internal	End Semester	
<b>Ability Enhancement Core Course (AECC)</b>						
U-COE-101	Communicative English-I	3	40	20	30	2
<b>Core Course (CC)</b>						
U-CFO-171	Computer Fundamentals and Office Automation	4	60	20	30	3
U-BDE-172	Basics of Digital Electronics	4	60	20	30	3
U-PIC-173	Programming in C	4	60	20	30	3
U-LOS-174	Linux Operating System	4	60	20	30	3
<b>Practical based on Core Course</b>						
U-LAC-175	Lab Course- I ( CF+OA)			--	50	2
U-LAC-176	Lab Course- II ( DE)			--	50	2
U-LAC-177	Lab Course- III (Prog. in C)			--	50	2
U-LAC-178	Lab Course- IV (LOS)			--	50	2
Total Marks and Credits (Sem-I)					450	22

**Semester: II**

Course Code	Course Title	Lectures		Marks		Credits
		Per Week	Total	Internal	End Semester	
<b>Ability Enhancement Core Course (AECC)</b>						
U-COE-201	Communicative English-II	3	40	20	30	2
<b>Core Course (CC)</b>						
U-DMA-271	Discrete Mathematics	4	60	20	30	3
U-DSA-272	Data Structures and Algorithms	4	60	20	30	3
U-CPR-273	C++ Programming	4	60	20	30	3
U-WPH-274	Web Programming with HTML and CSS	4	60	20	30	3
<b>Generic Elective (GE)</b>						
U-MOE-235	Moral Education	--	40	20	30	NCBC
<b>Practical based on Core Course</b>						
U-LAC-275	Lab Course- V (DM)			--	50	2
U-LAC-276	Lab Course- VI (DSA)			--	50	2
U-LAC-277	Lab Course- VII (C++)			--	50	2
U-LAC-278	Lab Course- VIII ( HTML+CSS )			--	50	2
Total Marks and Credits (Sem-II)					450	22

**Note:** Students can opt any online course such as SWAYAM, NPTEL, MOOCs will be given additional credits in total credits.

# **Semester - I**

**Rajarshi Shahu Mahavidyalaya(Autonomous), Latur**  
**U G First Year (Semester-I)**

---

**Course Title:** Communicative English - I  
**Max. Marks:** 50

**Course Code:** U-COE-101  
**Credits:** 2  
**Total Lectures:** 60

---

**Rajarshi Shahu Mahavidyalaya(Autonomous), Latur**  
**U G First Year (Semester-I)**

---

**Course Title:** Computer Fundamentals and Office Automation  
**Max. Marks:** 50

**Course Code:** U-CFO-171  
**Credits:** 3  
**Total Lectures:** 60

---

**Prerequisites:** No prior background required to learn the concepts of computer fundamentals. Students should complete the bridge course provided.

**Introduction:** This course deals with fundamentals of computer which includes generations and evolution of computers, input and output devices, primary and secondary storage devices, software etc. The focus is also given on office automation tools: MS-Word, MS-Excel and MS-PowerPoint which helps in official work at large extent.

### **Learning Objectives**

This course will enable students to:

1. Study the basics of Computer System
2. Learn how to configure computer devices
3. Understand working of operating system and other application software
4. Understand and recognize when to use each of the Microsoft Office programs to create professional and academic documents which will empower students for better employability skills.

### **Course Outcomes**

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

1. Analyze different computers, peripherals and software applications
  2. Explain the needs of hardware and software required for a computation task
  3. Use computer peripherals and memory efficiently
  4. Compare between various operating systems and their functions
  5. Use Microsoft Office applications such MS Word, Excel and PowerPoint in their day-to-day job, clerical and administration work, etc.
- 

## **SYLLABUS**

### **UNIT I**

**15**

Introduction to Computer System: Introduction, Characteristics of Computers, Block diagram of computer, Generations of computer, Classification of computers, **Data Representation within Computer:** Bit, Byte, Word, Codes: ASCII, EBCDIC, BCD, Decimal, Binary, Octal, Hexadecimal, **Input Devices:** Keyboard, Pointing devices: Mouse, Joystick, Touch Screen, Data Scanning devices, Digitizer, Electronic Card Reader, Voice Recognition devices, **Output Devices:** Monitor, Printer, Plotter, Screen image projector, voice response system

### **UNIT II**

**15**

**Introduction to Computer Peripherals:** Primary and Secondary storage devices, **Primary storage devices:** RAM, ROM, PROM, EPROM, EEPROM, Base Memory, Extended memory, expanded memory, cache memory, **Secondary Storage Devices:** Tape, FDD, HDD, CD ROM, Pen drive, **Computer Software:** Definition of software, Types of software, Compilers,

Interpreters, Assemblers, Linkers, Loaders **Introduction to Operating System:** Files and directories, Types of OS Types of Programming Languages- Machine Languages, Assembly Languages, High Level Languages, Translators- Assembler, Compiler, Interpreter

### UNIT III

17

**MS-Word:** Starting MS-Word, MS-Word Screen and its Components, **Working with Files, Working with Text:** Formatting, Moving, Copying and Pasting text, **Lists:** Bulleted and numbered lists, **Table Manipulations, Graphics:** Adding clip Art, add an image from a file, Spelling and Grammar, AutoCorrect, **Page formatting:** Header and footers, page numbers, Mail Merge

**MS-Excel:** Starting MS-Excel, Basics of Spreadsheet, **Working with worksheet:** Moving through cells, adding worksheets, rows and columns, Resizing rows and columns, selecting cells, Moving and copying cells, freezing panes. **Formatting cells:** Dates and times, Auto formatting. Formula and Functions. **Linking worksheets:** Sorting and Filling, Auto fill, **Graphics:** Adding clip art, add an image from a file, Charts: Using chart Wizard

### UNIT IV

13

**MS-PowerPoint:** Create a Presentation from a template, **Working with Slides:** Insert a new slide, applying a design template, changing slide layouts. **Slides:** Reordering slides, hide slides, Create a Custom slide show. Video and Audio effects, Adding clip art, adding an image from a file

### Reference Books:

1. Pradeep K. Sinha and Priti Sinha, Computer Fundamentals (Sixth Edition), BPB Publication
2. V Rajaraman, Fundamentals of Computer Science
3. Peter Norton's Introduction to Computers – Tata Mc Graw Hill
4. MS OFFICE 2000 Complete, BPB Publications, New Delhi
5. Ron Mansfield, Working with MS OFFICE, Tata Mc Graw Hill – 2000 Edition
6. John Walkenbach, Herb Tyson, Faithe Wempen, Cary N. Prague, Michael R. Groh, Peter G. Aitken, Michael R. Irwin, Gavin Powell, Lisa A. Bucki, Microsoft Office 2007 Bible, Wiley India Pvt. Ltd.
7. Lisa A. Bucki & John Walkenbach, Faithe Wempen, Michael Alexander & Dick Kusleika : Microsoft Office 2013 Bible-- Wiley India Pvt. Ltd.

**Rajarshi Shahu Mahavidyalaya(Autonomous), Latur**  
**U G First Year (Semester-I)**

---

**Course Title:** Basics of Digital Electronics  
**Max. Marks:** 50

**Course Code:** U-BDE-172  
**Credits:** 3  
**Total Lectures:**60

---

**Prerequisites:** Basic Electronics

**Introduction:** Digital electronics is a field of electronics involving the study of digital signals and the engineering of devices that use or produce them. Digital electronic circuits are usually made from large assemblies of logic gates, often packaged in integrated circuits. Computers are one of the most complex examples which makes use of complex circuits.

### **Learning Objectives**

This course will enable students to:

1. Understand common forms of number representation in logic circuits
2. Learn basic and universal logic gates
3. Understand the use of logic minimization methods and to solve the Boolean logic expressions.
4. Understand concepts of combinational and sequential circuits.

### **Course Outcomes**

At the end of the course, students will be able to:

1. Understand numerical information in different forms and Boolean algebra theorems
2. Apply Boolean logic and Karnaugh-Map to simplify the Boolean expressions
3. Implement and prove the truth tables of basic and universal logic gates
4. Apply Tabulation procedure to minimize Boolean expressions
5. Design and analyze combinational and sequential circuits

---

## **SYLLABUS**

### **Unit I**

**14**

Number system and codes: Binary number system, decimal number system, octal number system, hexadecimal number system. Bases inter conversions, Binary Subtraction using 1's and 2's complement method, Decimal subtraction using 9's and 10's complement method, Codes: BCD, GRAY, EXCESS-3.

### **Unit II**

**15**

Boolean algebra and logic systems: Boolean laws. Evaluation of Boolean expression, De Morgan's theorems and proof, simplification on Boolean expressions using Boolean laws, Basic gates (AND, OR, NOT): truth table, Definition, Boolean expression and symbols, universal gates (NAND, NOR): truth table, definition, Boolean expression and symbols, design of basic gates using NAND and NOR gates. Design of given Boolean expression using basic gates or NAND gate or NOR gate. XOR and XNOR gate (Definition, Boolean expression and symbols, truth table),



Minimization Techniques: Introduction, Minterms and Maxterms, K-Map, K-map for 2 variables  
K-map for 3 variables K-map for 4 variables)

### **Unit III**

**14**

Combination logic: Design procedure, design of half adder and full adder, half subtractor and full subtractor, Code converters: BCD to Excess 3 code, gray code, encoders (BCD to decimal), decoder (decimal to BCD), multiplexer (4:1 and 8:1), de-multiplexer (1:4 and 1:8).

### **Unit IV**

**17**

Sequential logic: Introduction, Flip-flops – SR flip flop, D flip flop, JK flip flop and T flip flop, Introduction, shift register: types and applications. Counters: synchronous and asynchronous counters (Up, down, up down), Introduction to microprocessors, Basic components of microprocessors.

### **Reference Books**

1. M. Moris Mano, Computer System Architecture, 2nd Edition, Prentice Hall of India.
2. Heuring and Jordan, Computer systems design and architecture, Pearson Education
3. William Stallings, Computer Organization and Architecture, Pearson Education 2003
4. Andrew S Tenenbaum, Structured Computer Organization, 3rd Edition, Prentice Hall of India (1990)
5. R.P. Jain, Modern Digital Electronics

**Rajarshi Shahu Mahavidyalaya(Autonomous), Latur**  
**U G First Year (Semester-I)**

---

**Course Title: Programming in C**  
**Max. Marks: 50**

**Course Code: U-PIC-173**  
**Credits:3**  
**Total Lectures: 60**

---

**Prerequisites:** Basic understanding of computer programming terminologies

**Introduction:** C is a procedural programming language. It was mainly developed as a system programming language to write an operating system. The main features of the C language include low-level memory access, a simple set of keywords, and own program style, these features make C language suitable for system programming like an operating system or compiler development.

### **Learning Objectives**

This course will enable students to:

1. Learn formulation of algorithm for a given problem and drawing flowchart for it
2. Understand the basic fundamentals and structure of C programming
3. Study various data types, arrays and functions in C
4. Understand input-output and, control and iterative statements in C
5. Develop the skills of C Programming
6. Improve the logical thinking skills

### **Course Outcomes**

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

1. Formulate an algorithm and draw flowchart for the given problem
  2. Implement the given algorithm in C
  3. Write programs using appropriate data types and control structures in C
  4. Understand and analyze various problem-solving methods
  5. Enhance their understanding of the logical flow of a program.
  6. Develop the skill of writing Algorithms and drawing Flowcharts
  7. Make a detailed study of the different decision-making structures and loop control structures
- 

## **SYLLABUS**

### **UNIT I**

**12**

**Basic Programming concepts-** Algorithm, Flowchart, **Overview of C:** Introduction, Importance of C, , Basic structure of C program, Sample 'C' programs, Executing a 'C' program, Data Types, C tokens, Keywords, Identifiers, Constants, Variables, Data types, Declaration of variables, Assigning values to variables, Defining symbolic constants, **Input and Output statements:** Input statements, Output statements, Reading character, Writing character, formatted input, formatted output statements, Simple Programs, Command line arguments

## UNIT II

15

**Operators and Expressions:** Arithmetic operators, Relational operators, Logical operators, Assignment operators, Increment and Decrement operators, Conditional operators, Bitwise operators, Special operators, Type Conversion in expressions, Operator precedence, Mathematical functions. Storage classes: auto, register, static, external

**Branching and Looping:** Simple 'if' statement, Simple, Nested, Ladder 'if-else' statement. 'Switch' statement, break, continue, goto, exit statement, 'while' statement, 'do-while' statement, 'for' statement, Simple programs on branching and looping.

## UNIT III

10

**Arrays:** Introduction, One dimensional, Two dimensional and Multidimensional arrays, Initialization of arrays, Handling of Character Strings: Declaring and Initializing string variables, reading string from terminal, writing string to screen, Arithmetic operations on characters, putting strings together, string handling functions: strlen, strcpy, strcat, strcmp.

## UNIT IV

13

**Functions:** Definition of function. Return values and their types, Function declaration, Function calls, Categories of function, Nesting of function, Recursion

**Structure and Union:** Introduction, Defining structure, declaring structure members and structure variables, arrays as structure, arrays within structure, Union. **Pointers:** Understanding Pointers, Accessing the address of variables, Declaring and initializing pointers, Accessing a variable through pointers.

### Reference Books

1. E. Balgurusamy, Programming in ANSI C (TMH)
2. Kamthane, Programming with ANSI and TURBO C (Pearson Education)
3. V. Rajaraman, Programming in C (PHI – EEE)
4. S. Byron Gottfried, Programming with C (TMH)
5. Kernighan & Ritchie, The C Programming Language (PHI)
6. Yashwant Kanitkar, Let us C
7. P.B. Kottur, Programming in C (Sapna Book House)

**Rajarshi Shahu Mahavidyalaya(Autonomous), Latur**  
**U G First Year (Semester-I)**

---

**Course Title: Linux Operating System**  
**Max. Marks: 50**

**Course Code: U-LOS-174**  
**Credits: 3**  
**Total Lectures: 60**

---

**Prerequisites:** No prior knowledge of Linux Operating System required to learn.

**Introduction:** Linux is one of the most popular open source Operating Systems. Linux refers to the family of Unix-like computer operating systems using the Linux kernel. Linux can be installed on a wide variety of computer hardware, ranging from mobile phones, tablet computers and video game consoles to mainframes and supercomputers.

### **Learning Objectives**

This course will enable students to:

1. Learn basics of open source Linux operating system
2. Demonstrate familiarity with Linux concepts and history
3. Learn file management and permission advance commands
4. Acquaint the practical knowledge of programming.
5. Learn file management and permission advance commands
6. Demonstrate the ability to use the Linux Shell for Command Processing
7. Understand Linux commands and write shell programming.
8. Grasp the concepts of User Management in Linux.

### **Course Outcomes**

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

1. Understand Linux installation, different Linux installation and Linux commands.
  2. Effectively use Linux Environment using shell, file system, scripts, filters and program development tools
  3. Perform file I/O management through commands and perform package management, storage management and failure recovery.
  4. Automate tasks and write simple programs using scripts
- 

## **SYLLABUS**

### **UNIT I**

**15**

**Introduction to Linux:** History of Linux, Advantages of Linux, Flavors of Linux, Linux Loader, Linux kernel, Some Basic Commands, Getting Help, Giving Multiple Commands, Aliases. **Files and File Organization:** Directory structure of Linux, Categories of Files: Hidden Files, File System: ext2, ext3, Path Names, Home Directory,

### **UNIT II**

**17**

File related commands: cat, cp, mv, rm, ls and Wild cards  
Filter commands: grep, wc, head, tail, sort  
Directory commands: pwd, cd, mkdir, rmdir  
Compression commands: zip, unzip, gzip, gunzip.  
Printer related commands: lpc, lpr, lpq, lprm

Working with permissions: File Attributes, Assigning file permission, Directory permission, Changing file permission, Changing the Owner of a File, Changing the group of a file, umask, Working System services and run levels

### **UNIT III**

**14**

Text Editors, vi Editor, Editing with vi: deleting character, text and line, **Shell Scripting:** Introduction to Shell, Basics of Shell Scripting: Shell Variables, Graphical Shells, Advantages of shell scripts, disadvantages of shell scripts

### **UNIT IV**

**14**

Performing system maintenance, **Communication commands:** write, wall, talk, mesg, motd, · Pre-login Message, **Managing software with RPM:** Installing, Uninstalling, Upgrading. Managing users and groups, Managing passwords.

### **Reference Books:**

1. Bill Ball, David Pitts, “Red Hat Linux 7 Unleashed”, Techmedia SAMS Publication
2. Evi Nemeth, Garth Snyder, Scott Seebass, Trent R. Hein, “UNIX System Administration Handbook”, Person Education Asia (LPE) (3rd Edition)
3. Bill Ball & Hoyt Dust, “Red hat Linux & fedora unleashed”

**Rajarshi Shahu Mahavidyalaya(Autonomous), Latur**  
**U G First Year (Semester-I)**

---

**Course Title: Lab Course-I (Computer Fundamentals and OA Tools)**

**Course Code: U-LAC-175**

**Max. Marks: 50**

**Credits: 2**

---

**Learning Objectives**

This course will enable students to:

1. Study the basics of Computer System
2. Learn how to configure computer devices
3. Understand working of operating system and other application software
4. Understand and recognize when to use each of the Microsoft Office programs to create professional and academic documents which will empower students for better employability skills.

**Course Outcomes**

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

1. Define working of computers and peripherals, types of software and languages
  2. Use computer peripherals and memory efficiently
  3. Troubleshoot the computer systems and use utility software
  4. Compare between various operating systems and their functions
  5. Use Microsoft Office applications such MS Word, Excel and PowerPoint in their day-to-day job, clerical and administration staff, support staff and their managers, business owners and entrepreneurs alike.
- 

**Proposed Practical List**

1. Study of elements of Windows OS
2. Study of Windows explorer: Creating and moving files and folders: a. Create a new folder and a file, b. Copy the created file to a new folder. c. Rename, Copy and Delete the created file and folder.
3. Study of control panel
4. Create an application and prepare a resume for applying a job in a company. Apply Page format and Paragraph format to the above document.
5. Create a news-paper document with at least 200 words in two column format and having an image.
6. Create a flowchart using proper shapes like ellipse, arrows, rectangle, and parallelogram.
7. Create a table using two columns: the left column contains all the short-cut keys and right side column contains the function of the short-cut keys
8. Create a letter to invite your friend for a function with at least 100 words and two paragraphs. The date must be in top-right corner.
9. Create a table "Student result" with following conditions. (a. The heading must contain, Sl. No., Name, Sub1, Sub2, Sub3, Total, Percentage and Result b. Use formulas for total and Percentage).

10. Create a sales table using the following data and draw the bar-graph to compare the sales of the three items for three years:

<b>Item</b>	<b>2019</b>	<b>2020</b>	<b>2021</b>
Item1	1000	1200	1100
Item2	950	1200	1150
Item3	1100	900	1250
Item4	1200	1250	800
Item5	1300	1000	1440

11. Create a power-point presentation for seminar with minimum 5 slides. (a. The first slide must contain the topic of the presentation and name of the presenter. b. Must contain at least one table. c. Last slide must contain “thank you”.)

12. Create a power-point presentation with minimum 10 slides having animation and images.

**Rajarshi Shahu Mahavidyalaya(Autonomous), Latur**  
**U G First Year (Semester-I)**

---

**Course Title: Lab Course-II (Basics of Digital Electronics)**

**Course Code: U-LAC-176**

**Max. Marks: 50**

**Credits: 2**

---

**Learning Objectives**

This course will enable students to:

1. Understand common forms of number representation in logic circuits
2. Learn basic and universal logic gates
3. Understand the use of logic minimization methods and to solve the Boolean logic expressions.
4. Understand concepts of combinational and sequential circuits.

**Course Outcomes**

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

1. Understand numerical information in different forms and Boolean algebra theorems
  2. Apply Boolean logic and Karnaugh-Map to simplify the Boolean expressions
  3. Implement and prove the truth tables of basic and universal logic gates
  4. Apply Tabulation procedure to minimize Boolean expressions
  5. Design and analyze combinational and sequential circuits
- 

**Proposed Practical List**

1. To perform and verify the truth tables of basic gates and derived gates
2. To perform and verify the truth tables of EX-OR and EX-NOR gate
3. To perform and verify LHS and RHS of Demorgan's Theorem I & II
4. To perform universal property of NAND gate
5. To perform universal property of NOR gate
6. To Implement and verify the truth table of any two Boolean equation
7. To perform and verify the truth table of half adder
8. To perform and verify the truth table of half subtractor
9. To perform and verify the truth table of multiplexer
10. To perform and verify the truth table of de-multiplexer
11. To perform and verify the truth table of encoder
12. To perform and verify the truth table of decoder



**Rajarshi Shahu Mahavidyalaya(Autonomous), Latur**  
**U G First Year (Semester-I)**

---

**Course Title: Lab Course-III (Programming in C)**

**Course Code: U-LAC-177**

**Max. Marks: 50**

**Credits: 2**

---

**Learning Objectives**

This course will enable students to:

1. Learn formulation of algorithm for a given problem and drawing flowchart for it
2. Understand the basic fundamentals and structure of C programming
3. Study various data types, arrays and functions in C
4. Understand input-output and, control and iterative statements in C
5. Develop the skills of C Programming
6. Improve the logical thinking skills

**Course Outcomes**

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

1. Formulate an algorithm and draw flowchart for the given problem
  2. Implement the given algorithm in C
  3. Write programs using appropriate data types and control structures in C
  4. Understand and analyze various problem-solving methods
  5. Enhance their understanding of the logical flow of a program.
  6. Develop the skill of writing Algorithms and drawing Flowcharts
  7. Make a detailed study of the different decision-making structures and loop control structures
- 

**Proposed Practical List**

1. Write a C program to find the area of a circle. Accept radius from user.
2. Write a C program to find the area of a rectangle. Input length and breadth from user.
3. Write a C program to calculate simple interest and compound interest.
4. Write a C program to convert temperature in Fahrenheit to Celsius and Celsius to Fahrenheit.
5. Write a C program to find the GCD and LCM of two integer numbers.
6. Write a C program to check whether the given integer is even or odd using if condition statement.
7. Write a C program to accept three integers and determine largest among them using if statement.
8. Write a C program to simulate a simple calculator with addition, subtraction, multiplication, division using switch case.
9. Write a C program to print number from 1 to 100 which are divisible by 7 and display their sum and count using for loop.
10. Write a C program to reverse a given integer number and check whether the number is palindrome or not using while loop.
11. Write a C program to check whether given number is prime or not using while loop.

12. Write a C program the pattern given below using nested for loop

```
      *
     * *
    * * *
   * * * *
  * * * * *
```

13. Write a C program to read N integers into an array and find sum of all numbers.

14. Write a C program to find the addition of two matrices.

15. Write a C program to calculate the factorial of a number using function.

---

**Rajarshi Shahu Mahavidyalaya(Autonomous), Latur**  
**U G First Year (Semester-I)**

---

**Course Title: Lab Course-IV (Linux Operating System)**  
**Max. Marks: 50**

**Course Code: U-LAC-178**  
**Credits: 2**

---

**Learning Objectives**

This course will enable students to:

1. Learn basics of open source Linux operating system
2. Demonstrate familiarity with Linux concepts and history
3. Learn file management and permission related commands
4. Acquaint the practical knowledge of Shell programming.
5. Demonstrate the ability to use the Linux Shell for Command Processing
6. Understand Linux commands and write shell programming.
7. Grasp the concepts of User Management in Linux.

**Course Outcomes**

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

1. Understand basics of open source operating system
  2. Perform Linux installation and run Linux commands
  3. Assign access permissions to file and directory
  4. Effectively use Linux Environment using shell, file system, scripts, filters and program development tools
  5. Automate tasks and write simple programs using scripts
- 

**Proposed Practical List**

1. Installation of Ubuntu/other version of Linux
2. Study of general purpose commands
3. Study of file related commands
4. Study of directory related commands
5. Study of file access permission and different types users in LINUX
6. Study of filter commands
7. Study of printer related commands
8. Study of communication commands
9. Study of Vi editor
10. Write a shell script program to find the Maximum three numbers
11. Write a shell script program for comparison of strings
12. Write a shell script to perform arithmetic operation using CASE
13. Write a shell script to calculate the factorial value of a number using shell script

# **Semester - II**

**Rajarshi Shahu Mahavidyalaya(Autonomous), Latur**  
**U G First Year (Semester-II)**

---

**Course Title:** Communicative English - II

**Max. Marks:** 50

**Course Code:** U-COE-201

**Credits:** 2

**Total Lectures:** 50

---

**Rajarshi Shahu Mahavidyalaya(Autonomous), Latur**  
**U G First Year (Semester-II)**

---

**Course Title: Discrete Mathematics**  
**Max. Marks: 50**

**Course Code: U-DMA-271**  
**Credits: 3**  
**Total Lectures: 60**

---

**Prerequisites:** Basic mathematics

**Introduction:**

Discrete mathematics is the branch of mathematics dealing with objects that can consider only distinct, separated values. This course includes the fundamental concepts of Sets, Relations and Functions, Mathematical Logic, Probability, Relations, Graph Theory and Trees.

**Learning Objectives**

This course will enable students to:

1. Understand the concepts of set theory, functions, logic and proof techniques
2. Build mathematical models to solve the real world problems by using appropriate methods
3. Enhance the problems solving skills in set theory, functions, relations and matrices
4. Learn graph theory which is to be used in software, networking and fabricating circuits

**Course Outcomes**

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

1. Demonstrate working knowledge of set notation and elementary set theory and logic  
Construct compound statements using logical connectives and verify the validity of conclusion using inference rules
2. Apply different properties of injections, surjections, bijections, compositions and inverse functions
3. Determine when a relation is reflexive, symmetric, antisymmetric and transitive, apply the properties of equivalence relations and partial ordering
4. Design and analyze various fundamentals of circuits
5. Construct graphs and trees which are widely in software

---

**SYLLABUS**

**UNIT I**

**16**

**Sets:** Definition and types of sets, Equal sets, subsets, Cardinal of sets, universal sets, Venn diagram, Set operations, Properties of set union and intersection, Cartesian product, **Relation:** Types of relation: Reflexive, Symmetric, Antisymmetric, Transitive, Equivalence and Partial order relation, **Function:** Domain and co-domain, range, Types of functions: Inverse function, one-to-one function, onto function, into function and composite function

**UNIT II**

**13**

**Mathematical Logic:** Propositions, Truth values and truth table, Logical connectives and compound statements, Statement pattern and logical equivalence, Tautology, contradiction, contingency

### UNIT III

14

**Matrices and Determinants:** Definition and types of matrices, Equality of Matrices and transpose of matrices, Algebra of matrices: addition, subtraction, scalar, multiplication of matrices, Adjoint of matrices, Inverse of matrices, Definition of determinant and numerical examples

### UNIT IV

17

**Graph Theory:** Definition and types of graph, Incidences and degree of vertices, Subgraphs, Isomorphism of graphs, Connected and disconnected graphs, Walks, paths and circuits, Directed graph, Hamiltonian path & graph, Operations on graphs **Tree:** Definition, Eccentricity, Centre of Tree, Binary Tree, Spanning tree, Cut sets and Cut vertices, Fundamental circuits, Edge connectivity and Vertex connectivity.

#### Reference Books:

1. C.L. Liu, Elements of Discreet Mathematics
2. Olympia Nicodemi, Discreet Mathematics
3. Alon Doerr and k. Levasieur, Mathematical Structures for Computer Science
4. Raghunathan, Nimkar & Solapurkar, A First Step in Graph Theory
5. Narsing Deo, Graphs theory with applications to Computer Science
6. Mittal and Agarwal, Basic Mathematics
7. Tremblay and Manohar, Discrete Mathematical Structures with Application to Computer Science, McGraw Hill Book Company.

**Rajarshi Shahu Mahavidyalaya(Autonomous), Latur**  
**U G First Year (Semester-II)**

---

**Course Title:** Data Structures and Algorithms

**Course Code:** U-DSA-272

**Max. Marks:** 50

**Credits:**3

**Total lectures:** 60

---

**Prerequisites:** Aware of at least one programming language

**Introduction:**

The data structure is the key component of Computer Science and is used in the areas of artificial intelligence, operating systems, graphics, etc. This course includes data structures such as queue, stack, linked list, graphs, trees, sorting algorithms, and searching algorithms.

**Learning Objectives**

This course will enable students to:

1. Learn the systematic way of solving problem
2. Identify and use appropriate data structures for a given problem with effective utilization of space and time.
3. Understand the different methods of organizing large amount of data
4. Implement solutions for specific problems efficiently
5. Apply the linear and nonlinear data structures
6. Analyze the complexities of different sorting techniques

**Course Outcomes**

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

1. Use well-organized data structures in solving various problems.
  2. Explore and understand the concepts of data structures and its significance in programming
  3. Differentiate the usage of various structures in problem solution.
  4. Implement algorithms to solve problems using appropriate data structures using C language.
- 

**SYLLABUS**

**UNIT I**

**15**

**Data structures:** Need of data structure, types of data structures, data structure operations, **Algorithms:** Definition and Characteristics, Time and space complexity, **Arrays:** Array terminology, characteristics of arrays, storage representation of 1D, 2D and Multi-dimensional arrays, operations on arrays

**UNIT II**

**15**

**Stacks:** Definition, Stack implementation, Operations on stack, Evaluation of arithmetic Expressions, Conversion of Expressions - Prefix, Infix and Postfix, Function Calling, Recursion-



direct & indirect recursion, Applications of Stacks, Queues: Definition, Operations on queues, Types of queues: Circular queue, Dequeue and Priority queues, Applications of queues

### **UNIT III**

**14**

Linked lists: Concept of linked list, Operations on Linked list, Types of linked list: Singly linked list, Circular linked list, Doubly linked list, Implementation of stack and queue using linked list, Applications of linked list.

### **UNIT IV**

**16**

Trees: Tree terminology, Binary trees, Traversing binary trees, Graphs: Graph terminologies, Graph representation: sequential and linked, Traversing a graph: Depth First Search and Breadth First Search **Sorting**: Selection sort, Insertion sort, Bubble sort, **Searching**: Linear search and Binary search

### **Reference Books**

1. Data Structure using C., A.M. Tanenbaum, Yecidyan lang
2. Fundamentals of Data Structures, Ellis Horowitz and Sartaj Sahni, Galgotia Publications
3. Introduction to Data Structures in C, Ashok N. Kamthane, Pearson Education
4. Theory and Problems of Data Structures, Seymour Lipschutz, Schaum's Outline Series

### **Online Resources:**

2. <http://nptel.ac.in/courses/106106127/>
3. <http://www.nptel.ac.in/courses/106102064>

# Rajarshi Shahu Mahavidyalaya(Autonomous), Latur

## U G First Year (Semester-II)

---

**Course Title:** C++ Programming

**Course Code:** U-CPR-273

**Max. Marks:** 50

**Credits:** 3

**Total lectures:** 60

---

**Prerequisites:** Fundamentals of programming

**Introduction:** C++ is a general-purpose, object-oriented programming language. It was created by Bjarne Stroustrup at Bell Labs circa 1980. C++ is very similar to C (invented by Dennis Ritchie in the early 1970s). C++ is so compatible with C that it will probably compile over 99% of C programs without changing a line of source code. Though C++ is a lot of well-structured and safer language than C as it OOPs based.

### Learning Objectives

This course will enable students to:

1. Explore the principles of Object Oriented Programming (OOP)
2. Understand object-oriented concepts such as data abstraction, encapsulation, inheritance, dynamic binding, and polymorphism
3. Use the object-oriented paradigm in program design
4. Lay a foundation for advanced programming
5. Provide programming insight using OOP constructs

### Course Outcomes

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

1. Analyze the strengths of object oriented programming
  2. Design and apply OOP concepts for effective programming
  3. Develop programming application using object oriented programming language C++
  4. Percept the utility and applicability of OOP
  5. Employ runtime error handling to develop a parallel processing application
- 

## SYLLABUS

### UNIT I

12

Introduction, Object Oriented Programming Paradigm, Basic concepts of OOP: Object, Class, Data Abstraction and Encapsulation, Inheritance, Polymorphism, Dynamic binding, Message Passing. **C++ Features:** The iostream class, Data types, Keywords, Comments, Variable declaration, I/O statement, Structure of C++ program, Scope resolution operator, new and delete operators. **Manipulators:** endl, setw, setprecision

## UNIT II

18

**Functions:** function declaration and definition, calling the function, pass by value, pass by reference, default arguments, inline functions. **Classes & objects:** Specifying a class, class member visibility: private, public, protected. Creating objects, Accessing class members, Defining member functions, Array of objects, friend function, constructors, default constructor, parameterized constructor, copy constructor, destructor.

## UNIT III

15

**Polymorphism:** Compile time and run time polymorphism, Function overloading, **Operator Overloading:** Concept of operator overloading, Defining operator overloading, Overloading unary operators, Overloading binary operators, Type conversions: basic to class type, Class to basic type

## UNIT IV

15

**Inheritance:** Derived class & Base class, Specifying the derived class, Accessing the base class members, Level of inheritance: Single inheritance, visibility of inherited members, Single inheritance, hybrid inheritance, multiple inheritance, Nesting of classes, Virtual base classes, **Virtual Function,** Pure virtual functions, Exception handling mechanism

### Reference Books

1. Herbert Schildt, Complete reference C++, Tata McGraw Hill
2. E. Balagurusami, Object Oriented Programming with C++, Fourth Edition, Tata Mc-Graw Hill
3. Ashok N. Kamthane, Object Oriented Programming with ANSI & Turbo C++, Pearson Education, 2006
4. John R. Hubbard, Theory and Problems of Programming with C++, Schaum's Outline Series, McGraw Hill
5. Bjarne Stroustrup, The C++ Programming Language, Third Edition, Addison-Wesley Publishing Company
6. Salaria, R. S., Object Oriented Programming Using C++, Fourth Edition, Khanna Book Publishing

# Rajarshi Shahu Mahavidyalaya(Autonomous), Latur

## U G First Year (Semester-II)

---

**Course Title: Web Programming with HTML and CSS**

**Course Code: U-WPH-274**

**Max. Marks: 50**

**Credits: 3**

**Total Lectures: 60**

---

**Prerequisites:** General familiarity with the way the web works is needed.

### **Introduction:**

HTML and CSS are the two most important languages for a new web developer to learn. This course will provide the basic knowledge & understanding of HTML, CSS and JavaScript using which students can design responsive web sites.

### **Learning Objectives**

This course will enable students to:

1. Learn the basics of website development
2. Acquaint the HTML tags and its attributes
3. Learn the CSS and Javascript
4. Apply CSS types for different web pages
5. Design an interactive web site using web technologies HTML, CSS and Javascript

### **Course Outcomes**

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

1. Describe different web technologies and application development issues and trends
  2. Distinguish between server-side and client-side web technologies
  3. Explain different components and technologies of World Wide Web as a platform
  4. Write valid and concise code for webpage
  5. Validate web form fields using JavaScript
  6. Demonstrate knowledge of artistic and design K2 components that are used in the creation of a web site.
  7. Develop an interactive web site using web technologies HTML, CSS and Javascript
- 

## SYLLABUS

### UNIT I

12

**Web Basics and HTML:** The Internet: Web clients, Web servers, Web browser, Basic Internet protocols, Client Server Architecture, HTTP Request and Response, URL, Phases of website development, Cross browser testing, how to publish a website, Introduction to HTML, Basic structure of an HTML document, **Markup Tags:** Basic HTML Tags, Physical style tags, Paragraphs, line breaks, HR, Heading levels, Center, Div, Span, Address, working with text, Lists

## UNIT II

18

**Working with hyperlink:** Hyperlink, Mailto anchor, **Working with Images:** Image file formats, <img> tag with its attributes, Images as background, Internal and External image, Image map: server side and client side image maps, Image as hyperlink, **Adding multimedia Elements:** Audio file formats, Adding audio in html document, Video file formats, Adding video in html document  
**Tables:** <table> tag with its attributes, Rowspan, colspan, **Frames:** Overview of frames, <frameset> tag and all attributes, Frame targeting, Floating frames, **Working with Forms and controls:** <form> tag with its attributes, Form controls, <input> tag with its attributes.

## UNIT III

18

Concept of CSS, Creating Style Sheet, Ways to insert CSS, CSS Properties, CSS Styling (Background, Text Format, Controlling Fonts), Working with block elements and objects, Working with Lists and Tables, CSS Id and Class, Box Model (Introduction, Border properties, Padding Properties, Margin properties), CSS Advanced (Grouping, Dimension, Display, Positioning, Floating, Align, Pseudo class, Navigation Bar, Image Sprites, Attribute selector), Color, Creating page Layout and Site Designs.

## UNIT IV

12

**Javascript:** Adding script to document, Working with local and global variable, Javascript, Data types, Operators, Control statements, Looping statements, **Document object model(DOM)**, DOM Objects (window, **navigator, history, location**), Predefined functions, math & string functions, **Array in Java scripts**, Event handling in Java script.

### Reference Books:

1. Monica D'Souza, Web Publishing
2. HTML 5 in simple steps, Kogent Learning Solutions Inc., Dreamtech Press
3. A beginner's guide to HTML, NCSA
4. Thomas Powell, HTML & CSS: The Complete Reference, Fifth Edition
5. Ivan Bayross, HTML and JavaScript
6. Mastering HTML, CSS & Javascript

### Reference websites:

1. [www.w3schools.com](http://www.w3schools.com)
2. [www.tutorialspoint.com](http://www.tutorialspoint.com)
3. <http://html.net>
4. <http://www.programmersneed.com>

# Rajarshi Shahu Mahavidyalaya(Autonomous), Latur

## U G First Year (Semester-II)

---

**Course Title: Lab Course-V (Discrete Mathematics)**  
**Max. Marks: 50**

**Course Code: U-LAC-275**  
**Credits: 2**

---

### Learning Objectives

This course will enable students to:

1. Understand the concepts of set theory, functions, logic and proof techniques
2. Build mathematical models to solve the real world problems by using appropriate methods
3. Enhance the problems solving skills in set theory, functions, relations and matrices
4. Learn graph theory which is to be used in software, networking and fabricating circuits

### Course Outcomes

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

1. Demonstrate working knowledge of set notation and elementary set theory and logic  
Construct compound statements using logical connectives and verify the validity of conclusion using inference rules
2. Apply different properties of injections, surjections, bijections, compositions and inverse functions
3. Determine when a relation is reflexive, symmetric, antisymmetric and transitive, apply the properties of equivalence relations and partial ordering
4. Design and analyze various fundamentals of circuits
5. Construct graphs and trees which are widely in software

---

### Proposed Practical List

1. MATLAB Overview
2. Study of Basics of MATLAB
3. Study of basic Commands
4. Study of Input output formatting
5. Study of various types of files
6. Performing Trivial Calculations
7. Creating and printing Simple 2D plot
8. Study of Publishing report in HTML and Word document
9. Creating matrix and indexing
10. Demonstrating types of matrices
11. Solving algebra of matrix
12. Study of inverse and determinant of matrix

# **Rajarshi Shahu Mahavidyalaya(Autonomous), Latur**

## **U G First Year (Semester-II)**

---

**Course Title: Lab Course-VI (Data Structures and Algorithms)**

**Course Code: U-LAC-276**

**Max. Marks: 50**

**Credits: 2**

---

### **Learning Objectives**

This course will enable students to:

1. Design and analyze linear and nonlinear data structures
2. Acquire programming skills to implement sorting and searching techniques
3. Identify and apply the suitable data structure for the given real world problem
4. Perform sorting and searching operations on data structure

### **Course Outcomes**

On completion of this course, students will able to:

1. Implement insert, delete, search, sort and traverse operations on array and linked list
2. Develop applications using stack and queue
3. Apply nonlinear data structures to solve computing problems
4. Implement optimized sorting technique for a given data set
5. Implement hashing techniques to perform dictionary operations

---

### **Proposed Practical List**

1. Implementation of insertion and deletion algorithms of stack
2. Implementation of simple queue using an array.
3. Implementation of:
  - a. Infix to Postfix conversion
  - b. Postfix Evaluation
4. Implementation of the following
  - a. Simple queue
  - b. Circular queue
5. Implementation of Linked list algorithm for insertion and deletion of an item from list
6. Implementation of Binary Tree traversal algorithms
7. Recursive implementation of
  - a. Factorial
  - b. Fibonacci
  - c. Tower of Hanoi
8. Implementation of Linear Search algorithm to find an item in a list.
9. Implementation of Binary search algorithm to find an item in an ordered list.
10. Implementation Sorting Algorithms
  - a. Bubble sort
  - b. Insertion sort
  - c. Selection sort

**Rajarshi Shahu Mahavidyalaya(Autonomous), Latur**  
**U G First Year (Semester-II)**

---

**Course Title: Lab Course-VII (C++ Programming)**

**Course Code: U-LAC-277**

**Max. Marks: 50**

**Credits: 2**

---

### **Learning Objectives**

This course will enable students to:

1. Explore the principles of Object Oriented Programming (OOP)
2. Understand object-oriented concepts such as data abstraction, encapsulation, inheritance, dynamic binding, and polymorphism
3. Use the object-oriented paradigm in program design
4. Lay a foundation for advanced programming
5. Provide programming insight using OOP constructs

### **Course Outcomes**

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

1. Analyze the strengths of object oriented programming
  2. Design and apply OOP concepts for effective programming
  3. Develop programming application using object oriented programming language C++
  4. Percept the utility and applicability of OOP
  5. Employ runtime error handling to develop a parallel processing application
- 

### **Proposed Practical List**

1. Program to demonstrate I/O statements and manipulators
2. Program for scope resolution operator
3. Write a program using area () function that returns the area of a circle with given radius.  
float area (float r)
4. Write a program using min () function that returns the smallest of two integers  
int min (int x, int y)
5. Program to demonstrate class and object concept
6. Program to demonstrate constructor and its types
7. Program to demonstrate destructor
8. Program to find area and circumference of rectangle and triangle using function overloading
9. Program to demonstrate operator overloading concept
10. Program to demonstrate single inheritance
11. Program to demonstrate multiple inheritance
12. Program to demonstrate virtual function
13. Program to demonstrate exception handling mechanism



**Rajarshi Shahu Mahavidyalaya(Autonomous), Latur**  
**U G First Year (Semester-II)**

---

**Course Title: Lab Course-VIII (Web Programming with HTML and CSS)**

**Course Code: U-LAC-278**

**Max. Marks: 50**

**Credits: 2**

---

**Learning Objectives**

This course will enable students to:

1. Learn the basics of website development
2. Acquaint the HTML tags and its attributes
3. Learn the CSS and Javascript
4. Apply CSS types for different web pages
5. Design an interactive web site using web technologies HTML, CSS and Javascript

**Course Outcomes**

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

1. Describe different web technologies used for web application development
  2. Distinguish between server-side and client-side web technologies
  3. Write valid and concise code for webpage
  4. Validate web form fields using JavaScript
  5. Develop an interactive web site using web technologies HTML, CSS and Javascript
- 

**Proposed Practical List:**

1. Prepare a survey document of five websites and find out prerequisite for designing it.
2. Design web pages for your college containing a description of the courses, departments, faculties, library etc, Make use of <a> and list tags.
3. Create your class timetable using <table> tag.
4. Create Student feedback form (use textbox, text area, checkbox, radio button, select box etc.)
5. Create a web page using frame. Divide the page into two parts with Navigation links on left hand side of page (width=20%) and content page on right hand side of page (width = 80%). On clicking the navigation Links corresponding content must be shown on the right hand side.
6. Create your resume using HTML tags and make use of colors, text, link, size and also other tags you studied.
7. Design a web page of your home town with an attractive background color, text color, an Image, font etc. (use internal CSS).
8. Use Inline CSS to format your resume that you have created in experiment-6.
9. Use External CSS to format your class timetable as you have created in experiment-3.
10. Use External, Internal, and Inline CSS to format college web page that you have created in experiment-2.
11. Write a JavaScript program to display system date.
12. Write a JavaScript program to develop simple calculator for arithmetic operations.
13. Write a JavaScript program to determine whether a given year is a leap year or not.
14. Write a JavaScript program to convert temperatures to and from celsius, Fahrenheit.
15. Design sign-up form to validate username, password using Java script.

**Rajarshi Shahu Mahavidyalaya(Autonomous), Latur**  
**U G First Year (Semester-II)**

---

**Course Title: Lab Course-VII (C++ Programming)**

**Course Code: U-LAC-278**

**Max. Marks: 50**

**Credits: 2**

---

**Learning Objectives**

This course will enable students to:

6. Explore the principles of Object Oriented Programming (OOP)
7. Understand object-oriented concepts such as data abstraction, encapsulation, inheritance, dynamic binding, and polymorphism
8. Use the object-oriented paradigm in program design
9. Lay a foundation for advanced programming
10. Provide programming insight using OOP constructs

**Course Outcomes**

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

6. Analyze the strengths of object oriented programming
  7. Design and apply OOP concepts for effective programming
  8. Develop programming application using object oriented programming language C++
  9. Percept the utility and applicability of OOP
  10. Employ runtime error handling to develop a parallel processing application
- 

**Proposed Practical List**

1. Program to demonstrate I/O statements and manipulators
2. Program for scope resolution operator
3. Write a program using area () function that returns the area of a circle with given radius.  
float area (float r)
4. Write a program using min () function that returns the smallest of two integers  
int min (int x, int y)
5. Program to demonstrate class and object concept
6. Program to demonstrate constructor and its types
7. Program to demonstrate destructor
8. Program to find area and circumference of rectangle and triangle using function overloading
9. Program to demonstrate operator overloading concept
10. Program to demonstrate single inheritance
11. Program to demonstrate multiple inheritance
12. Program to demonstrate virtual function
13. Program to demonstrate exception handling mechanism