



**SOLAPUR UNIVERSITY, SOLAPUR**

**FACULTY OF ENGINEERING & TECHNOLOGY**

**COMPUTER SCIENCE AND ENGINEERING**

**Syllabus Structure for**

**S.E. (Computer Science & Engineering) w.e.f. Acad. Year 2017-18**

**T.E. (Computer Science & Engineering) w.e.f. Acad. Year 2018-19**

**B.E. (Computer Science & Engineering) w.e.f. Acad. Year 2019-20**

**Choice Based Credit System**



**SOLAPUR UNIVERSITY, SOLAPUR**  
**FACULTY OF ENGINEERING & TECHNOLOGY**  
**Computer Science & Engineering**

**Programme Educational Objectives and Outcomes**

***A. Program Educational Objectives***

1. To make students competent for professional career in Computers, IT & allied fields.
2. To build strong fundamental knowledge amongst student to pursue higher education and continue professional development in Computers, IT & other fields
3. To imbibe professional ethics, develop team spirit and effective communication skills to be successful leaders and managers with a holistic approach.
4. To nurture students to be sensitive to ethical, societal & environmental issues while conducting their professional work.

***B. Program Outcomes***

Engineering Graduate will be able to –

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

- 8. Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
- 9. Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
- 10. Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
- 11. Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
- 12. Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.



**SOLAPUR UNIVERSITY, SOLAPUR**  
**Faculty of Engineering & Technology**  
**Second Year Computer Science and Engineering**

*Choice Based Credit System Syllabus Structure of S.E.Computer Science and Engineering W.E.F. 2017-2018 Semester I*

Course Code	Theory Course Name	Hrs./week			Credits	Examination Scheme				
		L	T	P		ISE	ESE	ICA	Total	
CS 211	Applied Mathematics - I	3	1	-	4	30	70	25	125	
CS 212	Discrete Mathematical Structures	3	1	-	4	30	70	25	125	
CS 213	Data Communication	3	-	-	3	30	70	-	100	
CS 214	Digital Techniques	3	-	-	3	30	70	-	100	
CS 215	Computer Graphics	3	-	-	3	30	70	-	100	
CS 216	Advanced C Concepts	3	-	-	3	--	--	-	--	
	<b>Sub Total</b>	18	2	-	20	150	350	50	550	
ENV 21	<b>Environmental Studies</b>	1								
	<b>Laboratory</b>									
							<b>ESE</b>			
							<b>POE</b>	<b>OE</b>		
CS 213	Data Communication	-	-	2	1	-	50	--	25	75
CS 214	Digital Techniques	-	-	2	1	-	50	-	25	75
CS 215	Computer Graphics	--	--	2	1	--		--	25	25
CS 216	Advanced C Concepts	-	-	4	2	-	50	-	25	75
	<b>Sub Total</b>	--	-	10	5	-	150		100	250
	<b>Grand Total</b>	<b>18</b>	<b>2</b>	<b>10</b>	<b>25</b>	<b>150</b>	<b>500</b>	<b>150</b>	<b>800</b>	

- Abbreviations: L- Lectures, P –Practical, T- Tutorial, ISE- In Semester Exam, ESE End Semester Exam, ICA- Internal Continuous Assessment, ESE - University Examination (Theory &/ POE &/Oral examination)



**SOLAPUR UNIVERSITY, SOLAPUR**  
**Faculty of Engineering & Technology**  
**Second Year Computer Science and Engineering**

*Choice Based Credit System Structure of S.E. Computer Science and Engineering W.E.F. 2017-2018 Semester II*

Course Code	Theory Course Name	Hrs./week			Credits	Examination Scheme				
			T	P		ISE	ESE	ICA	Total	
CS 221	Applied Mathematics – II	3	1	–	4	30	70	25	125	
CS 222	Theory of Computation	3	1	–	4	30	70	25	125	
CS 223	Microprocessors	3	–	–	3	30	70	–	100	
CS 224	Data Structures	3	–	–	3	30	70	–	100	
CS 225	Computer Networks	3	–	–	3	30	70	–	100	
CS 226	Object Oriented Programming through C++	3	–	–	3	–	–	–	–	
	<b>Sub Total</b>	18	2	–	20	150	350	50	550	
ENV22	Environmental Studies	1								
	Laboratory									
							<b>ESE</b>			
							<b>POE</b>	<b>OE</b>		
CS 223	Microprocessors	–	–	2	1	–	50	–	25	75
CS 224	Data Structures	–	–	4	2	–	50	–	25	75
CS 225	Computer Networks	–	–	2	1	–	–	–	25	25
CS 226	Object Oriented Programming through C++	–	–	2	1	–	50	–	25	75
	<b>Sub Total</b>		–	10	5	–	150	–	100	250
	<b>Grand Total</b>	<b>18</b>	<b>2</b>	<b>10</b>	<b>25</b>	<b>150</b>	<b>500</b>	<b>150</b>	<b>800</b>	

- Abbreviations: L- Lectures, P –Practical, T- Tutorial, ISE- In Semester Exam, ESE - End Semester Exam, ICA- Internal Continuous Assessment, ESE - University Examination (Theory &/ POE &/Oral examination)

**Note –**

- Batch size for the practical /tutorial shall be of 20 students. On forming the batches, if the strength of remaining student exceeds 9, then a new batch shall be formed.
- Vocational Training (evaluated at B.E. Part-I) of minimum 15 days shall be completed in any vacation after S.E. Part-II but before B.E. Part-I & the report shall be submitted and evaluated in B.E. Part-I
- Term work assessment shall be a continuous process based on student’s performance in – class tests, assignments, homework, subject seminars, quizzes, laboratory books and their interaction and attendance for theory and lab sessions as applicable.



**Solapur University, Solapur**  
**S.E. ( Computer Science and Engineering) Semester-I**  
**CS 211 APPLIED MATHEMATICS-I**

**Teaching Scheme:**  
**Scheme**

**Lectures- 3 Hours / week, 3 Credits**

**Tutorial - 1 Hours / week, 1 Credit**

**Examination**

**ESE- 70 Marks**

**ISE - 30 Marks**

**ICA- 25 Marks**

---

This course includes mathematical theory and concepts required by the computer engineer. The course consists of linear differential equations which can be used for mathematical model which are appearing in computer engineering, where these variables are dynamically related. This course introduces Z- transform which provide a mathematical framework for a series of mathematical conversions that are useful for digital filters. Laplace transforms is another powerful mathematical tool for engineering problems in Computer Science and Engineering. This course also introduces Fourier series, which plays an important role in designing, and analyzing communication system. This course also introduces fundamentals of probability distributions which are useful for digital communication. This course introduces Queuing systems which are prevalent throughout society. There are applications of queuing theory in several disciplines.

---

**Course Prerequisite:**

Fundamentals of trigonometry, method of finding roots of algebraic equations, differentiation, integration, partial fraction, sum of sequence and methods of solving definite integrations, basics of statistics and probability theory

---

**Course Objectives:**

1. To introduce to student method of solving higher order linear differential equations
2. To introduce to student Laplace and inverse Laplace transforms
3. To introduce to student Fourier series of periodic functions
4. To make student understand Z transform and its properties
5. To introduce to student various probability distributions
6. To introduce to student Queuing theory and its model.

---

**Course Outcomes:**

1. Student can solve higher order linear differential equation with constant coefficient
2. Student can apply Laplace and inverse Laplace transforms for solving linear differential equations.
3. Student can express a function in terms of sine's and cosines components so as to model simple periodic functions.
4. Student can solve problems on Z transform and explain its properties
5. Student can find the relation between two variables for the given data using regression
6. Student can sketch and explain various probability distribution functions

7. Student can solve problems based on queuing theory.

## SECTION – I

**Unit 1: Linear differential equations with constant coefficients:** No. of lectures-07

Basic definition, differential operator, complimentary functions, particular integral, Shortcut methods for standard functions like  $e^{ax}$ ,  $\sin(ax + b)$ ,  $\cos(ax + b)$ ,  $x^m$ ,  $e^{ax}V$  and  $xV$ , particular integral by general method (without method of variation of parameters) for other functions.

**Unit 2: Z-Transform:** No. of lectures-05

Introduction, Z-Transform of standard sequence, properties of Z-transform – linearity, change of scale, shifting property, multiplication by k, division by k, inverse Z-transform –power series method, partial fraction method

**Unit 3: Laplace transform:** No. of lectures-09

Definition, Laplace transform of standard functions, properties- first shifting, change of scale, multiplication of power t and division by t, Laplace transform of derivative and integral, Laplace transform of periodic functions, unit step functions and unit impulse functions, properties of inverse Laplace transforms- linear property, first shifting theorem, partial fraction, inverse transform of logarithmic & inverse trigonometric functions and convolution theorem, solution of differential equations by Laplace transform.

## SECTION-II

**Unit 4: Fourier series:** No. of lectures-07

Introduction, Definition, Euler's formula, Fourier series of periodic functions with period  $2\pi$  and  $2L$ , Dirichlet's theorem (only statement), even and odd functions, half range sine and cosine series.

**Unit 5: Statistics and probability:** No. of lectures-08

Coefficient of correlation and lines of regression of bivariate data, random variable, Binomial, Poisson, Normal distribution

**Unit 6: Queuing Theory:** No. of lectures-06

Introduction, Queuing system, Distributions in Queuing systems, Kendall's notation, classification of queuing models, M/M/1:  $\infty/\infty$  models, M/M/1:  $N/\infty$  models.

- **Internal Continuous Assessment (ICA):**

*ICA shall consist of minimum six to eight assignments based on entire curriculum*

- **Text books:**

1. A textbook of Applied Mathematics Vol. II and Vol. III, J.N. and P.N. Wartikar, Vidyarthi Grah Prakashan, Pune.
2. Higher Engineering Mathematics, Dr.B.S.Grewal, Khanna Publications, Delhi.
3. A Textbook of Applied Mathematics, N.P. Bali, Ashok Saxena and N.Ch. S.N. Iyengar, Laxmi Publications, Delhi.
4. Advanced Engineering Mathematics, Kreyzig-John Wiley & SMS, New York.

- **Reference Books:**

1. Advanced Engineering Mathematics, Peter O'Neil , Cengage Learning.
2. Engineering Mathematics, Srimanta Pal, Subodh Chandra Bhunia, Oxford University Press





**Solapur University, Solapur**  
**S.E. (Computer Science & Engineering)**  
**Semester-I**

**CS 212 - DISCRETE MATHEMATICAL STRUCTURES**

**Teaching Scheme**

Lectures– 3Hours/week, 3 Credits  
Tutorial – 1 Hour/week, 1 Credits

**Examination Scheme**

ESE – 70Marks  
ISE – 30Marks  
ICA - 25 Marks

---

**Introduction:**

This course introduces discrete mathematics which deals with fundamentals of mathematical reasoning and set theory. The course also introduces theoretical and mathematical aspects of relations, functions, algebraic system & Boolean algebra.

---

**Course Prerequisite:** Student shall have knowledge of basic mathematics.

---

**Course Objectives:**

1. To study basic connectives, equivalent formulas and normal forms.
  2. To understand set theory and relations with examples.
  3. To understand functions and types of function through examples.
  4. To make student understand definition and types of algebraic systems, lattices, Boolean algebra and their applications.
- 

**Course Outcomes:**

Students will be able to:

1. To solve problems on connectives, Tautological implications and equivalence of formulas.
  2. Students will be able to solve problems on set theory and set algebra.
  3. To correlate the concept of relation and function.
  4. To describe and utilize the concepts of algebraic system, groups.
  5. To describe Boolean algebra and properties of Boolean algebra.
- 

**SECTION-I**

<b>UNIT-1</b>	<b>Mathematical logic</b>	<b>6 hrs</b>
	Introduction, statements and Notation, Connectives - negation, conjunction, disjunction, conditional, bi conditional, statement formulas and truth tables, well-formed formulas, Tautologies, Equivalence of formulas, Duality law, Tautological implications, functionally complete sets of connectives, other connectives.	
<b>UNIT-2</b>	<b>Representation of expressions</b>	<b>4 hrs</b>
	Normal & Principle normal forms, completely parenthesized infix & polish notations, Theory of inference for statement calculus.	
<b>UNIT-3</b>	<b>Set theory</b>	<b>4 hrs</b>
	Basic concepts of set theory, types of operations on sets, ordered pairs, Cartesian product.	

<b>UNIT - 4</b>	<b>Relations</b>	<b>6 hrs</b>
	Relations, Properties of binary relations, Matrix and graph representation, Partition and covering of set, Equivalence relation, Composition, POSET and Hasse diagram.	

## SECTION II

<b>UNIT- 5</b>	<b>Functions</b>	<b>4 hrs</b>
	Function -types, Composition of functions, Inverse functions.	
<b>UNIT- 6</b>	<b>Algebraic systems</b>	<b>4 hrs</b>
	Algebraic systems, semi groups and monoids, properties and example.	
<b>UNIT- 7</b>	<b>Groups</b>	<b>4 hrs</b>
	Polish expressions and their compilation, Groups, group codes.	
<b>UNIT - 8</b>	<b>Lattices and Boolean algebra</b>	<b>7 hrs</b>
	Lattice as POSETs, definition, examples and Properties, Special lattices, Boolean algebra definition and examples, Boolean functions.	

### Internal Continuous Assessment (ICA):

In tutorial session, students of different batches should be assigned exercise problems and should be guided for the solution. Minimum one tutorial per unit is expected.

*ICA shall consist of minimum ten assignments from the below list.*

1. Problems on Statements and Notations, Problems on Construction of truth tables for the given formulas.
2. Problems on well-formed formulas and identify which one is tautology, contradiction or contingency.
3. Problems on equivalence of formulas, tautological implications, on other connectives (ex. conversion of given formula into NAND and NOR only).
4. Problems on Normal Forms (DNF, CNF).
5. Problems on PDNF and PCNF (with and without constructing the truth table).
6. Problems on conversion of infix to prefix and postfix by constructing a tree (infix and polish notations).
7. Problems on determining the validity of the argument (Theory of inference for statement calculus).
8. Problems on Basic concepts of Set Theory, types of operations on sets & on Cartesian product.
9. Problems on properties of relations, conversion of a relation into Matrix and Graph.

10. Problems on Partition and Covering of set, equivalence relation, composition of relation.
11. Problems on POSET and Hasse Diagram.
12. Problems on types of functions, composition of function and inverse function.
13. Problems on Algebraic System (Semigroup, Monoid, Group and Abelian group, etc).
14. Problems on conversion of infix to reverse polish (infix to postfix).

**Text Books:**

1. Discrete mathematical structures with applications to computer science -- J. P. Tremblay & R. Manohar (MGH International)

**Reference Books:**

1. Discrete Mathematics with combinatorics and graph theory- S. SNTHA (CENGAGE Learning)
2. Discrete Mathematical Structures – Bernard Kolman, Robert C. Busby (Pearson Education)
3. Discrete mathematics -- Liu (MGH)
4. Theory and problems in Abstract algebra -- Schaums outline series (MGH)
5. Discrete Mathematical Structures- Y N Singh (WILEY)



## Solapur University, Solapur

S.E. (Computer Science & Engineering)

Semester-I

CS 213 – DATA COMMUNICATION

### Teaching Scheme

Lectures– 3 Hours/week, 3 Credits

Practical – 2 Hour/week, 1 Credits

### Examination Scheme

ESE – 70Marks

ISE – 30Marks

ICA - 25 Marks

POE – 50 Marks

### Introduction:

This course introduces Data Communication Fundamentals and Computer Networks basic knowledge. It also covers the details of Physical Layer, Data Link Layer and Network Layer design issues.

**Course Prerequisite:** Students should have knowledge of Computer Systems and basics of C programming language.

### Course Objectives:

1. To study Data Communication Fundamentals such as Data, Signals, Transmission medias
2. To understand Uses of Computer Network, OSI Reference model and TCP/IP
3. To understand Data Link layer in detail
4. To study Network Layer and its protocols/algorithms

### Course Outcomes:

Students will be able to:

1. Get acquainted with Data Communication basics.
2. To describe the use of OSI reference model and TCP/IP
3. To correlate the concept of Data Link Layer functions
4. To describe and utilize different protocols used in Network Layer

## SECTION-I

UNIT-1	Data Communication Fundamentals	7 hrs
	Concepts & Terminology, Analog & Digital data transmission, Transmission impairments, Channel capacity, guided transmission media, Digital data to digital signal encoding, Asynchronous and Synchronous Transmission.	
UNIT-2	Reference Models	7 hrs
	Uses of Computer network, Network hardware, Network software, Types of Network Topologies, OSI reference model, TCP/IP protocol, ATM reference model.	
UNIT-3	Data Link Layer	6 hrs
	DLL design issues, Error detection & correction, Elementary DLL protocols, Sliding window protocols.	

## SECTION II

<b>UNIT- 4</b>	<b>Medium Access Control</b>	<b>6 hrs</b>
	Channel allocation problems, Multiple access protocol: ALOHA, CSMA, CSMA/CD, Collision free protocols, Limited contention protocols.	
<b>UNIT- 5</b>	<b>IEEE standards</b>	<b>6 hrs</b>
	IEEE standards 802.3, 802.4, 802.5 & 02.6(DQDB), Bridges, Switches, Routers	
<b>UNIT- 6</b>	<b>Network Layer</b>	<b>7 hrs</b>
	Network layer design issues, Routing algorithms: shortest path routing, flooding, flow-based routing, distance vector routing, link state routing, hierarchical routing, Congestion control algorithms, Internetwork	

### ICA:

Student should perform minimum 8 experiments based on the following guidelines.

1. Implementation of simplex, half duplex and full-duplex using RS 232 C (9 pin) standard and bioscom function.
2. File transfer using RS 232C std. and bioscom function.
3. Simulation of different Framing methods. (Character count, starting and ending flag etc)
4. Implement error detection method – CRC
5. Implement error detection and correction method - Hamming code.
6. Implement a program for simulation of sliding window protocols.
7. Implementation of IEEE 802.3, 802.4 and 802.5 frame formats and conversion of 802.x frame into 802.y frame.
8. Implementation of Shortest path routing algorithm.
9. Implementation of Flow – based routing algorithm.
10. Given the IP address find out class, subnetmask, netid and hostid.

### Text Books:

1. Data & Computer Communication ( Unit 1)  
William Stallings. (seventh edition) PHI publications.
2. Computer Networks (Unit 2, 3, 4, 5,6)  
Andrew S. Tanenbaum (third edition) PHI publications.

### Reference Books:

1. Computer Networks (Principles, Technologies and Protocols for network design) - Natalia Olifer, Victor Olifer (Wiley Publications)



**Solapur University, Solapur**  
**S.E. (Computer Science & Engineering)**  
**Semester-I**  
**CS 214 - DIGITAL TECHNIQUES**

**Teaching Scheme**

Lectures – 3 Hours/week, 3 Credits  
Practical – 2 Hour/week, 1 Credit

**Examination Scheme**

ESE – 70 Marks  
ISE – 30 Marks  
ICA – 25 Marks  
POE – 50 Marks

---

**Introduction:**

This course provides a thorough introduction to the Digital Logic Circuits and the basic building blocks used in digital systems. The course also introduces Hardware Description Language. It starts with a discussion of combinational logic circuits: logic gates, minimization techniques, arithmetic circuits, data processing circuits. The second part of the course deals with sequential circuits: flip-flops, registers and counters.

---

**Course Prerequisite:** Student has completed a Course in Basic Electronics and has an adept knowledge of electronic components and fundamental of digital electronics.

---

**Course Objectives:**

1. To study various number systems and Boolean algebra.
2. To introduce concept of digital principles, digital logic circuits and its advantages.
3. To develop designing and implementation skills of combinational logic circuits and sequential circuits.
4. To design and simulate digital circuits using Hardware Description Language (HDL).

---

**Course Outcome:**

Students will be able to

1. Apply the concepts of various number systems, Boolean algebra to solve relevant problems.
2. Design digital logic circuits.
3. Design and implement combinational logic circuits and sequential circuits.
4. Simulate different logic circuits using Hardware Description Language (HDL).

---

**SECTION I**

<b>UNIT-1</b>	<b>Digital Principles and Number Systems</b>	<b>7hrs</b>
	Introduction, Number Systems, Binary Number System, Signed Binary Numbers, Binary Arithmetic, 2's Complement Arithmetic, Octal Number System, Hexadecimal Number System, Codes. Introduction, Digital Signals, Basic Digital Circuits, NAND and NOR Operations, Exclusive-OR and Exclusive-NOR Operations, Boolean algebra, Examples of IC Gates.	

<b>UNIT-2</b>	<b>Combinational Logic Design</b>	<b>7 hrs</b>
	Introduction to Standard representation of Logical function, K-map representation, Simplification using k-map up to 4 variables, Minimization of logical function specified in minterm / maxterms, Don't care condition, AND / OR/ XOR function using NOR / NAND gates, Implementation of SOP & POS expression using NAND & NOR gate respectively, Arithmetic Circuits :Half adder, Full adder, Half subtractor, Full subtractor.	
<b>UNIT-3</b>	<b>Combinational Logic Design using MSI Circuits</b>	<b>6 hrs</b>
	Data selector / multiplexer, MUX as logic function, Decoder / Demultiplexes, application of MUX/ DEMUX using IC – 74151, 74154, 74148, problems based on MSI circuits, decoder / driver for 7 segment display using 7447. Digital Comparator, Parity Checker, Arithmetic Logic Unit.	

## SECTION II

<b>UNIT-4</b>	<b>Sequential Logic Circuit</b>	<b>7 hrs</b>
	Introduction to flip-flops, S-R, J-K, D-Flip-flop & T-flip-flop, Excitation Tables for Flip flops, Basic register, Shift register, Asynchronous counters, Up-down counter, MOD counter, Introduction to synchronous counter.	
<b>UNIT-2</b>	<b>Introduction to Verilog HDL</b>	<b>6 hrs</b>
	Introduction to Verilog HDL, HDL Implementation Models: Dataflow Modeling, Behavioral Modeling, HDL Implementation of Arithmetic circuits: Half Adder, Full Adder, Half subtractor, Full subtractor. 4-bit ripple carry adder.	
<b>UNIT-3</b>	<b>Digital Design with Verilog HDL</b>	<b>6 hrs</b>
	HDL Implementation of Data Processing Circuits: 2:1, 4:1, 8:1 Multiplexer, BUS Representation in HDL, HDL Implementation of Flip-Flops, Register, Counters.	

### ICA (Internal Continuous Assessment):

Student should perform 8 to 9 Experiments **using Trainer Kit / Bread Board** and **Synthesis tool: Xilinx ISE.**

1. a) Verification of truth table of basic and universal logic gates.  
b) Implementation of Boolean functions using Basic and Universal Gates.
2. Implementation of reduced Boolean functions (K – map technique) from Assignment No 1 using basic and universal gates.
3. Implementation of half adder and full adder using Basic Gates.
4. a) Design and Implement a 4:1 multiplexer using 74153 IC.  
b) Design and Implement a 8:1 multiplexer using 74151 IC.
5. Implementation of flip flops using NAND/NOR gates: S-R Flip flop, D flip flop.
6. Implementation of Mod 10 (Decade) Counter & Mod 5 counters using IC 7490.

7. a) Write Verilog code to realize all the logic gates, Simulate and verify it's working.  
b) Design the Half subtractor and full Subtractor using Verilog Code, Simulate and verify its working.
8. a) Design a 4:1 & 8:1 multiplexer circuit using Verilog Code, Simulate and verify it's working.  
b) Design a 1:8 Demultiplexer circuit using Verilog Code, Simulate and verify it's working.
9. Design the following flip-flops, SR, D, JK, T using Verilog Code, Simulate and verify its working.
10. Design 4 bit Synchronous and Asynchronous binary using Verilog Code, Simulate and verify its working.

**Text Book:**

1. R. P. Jain: Modern Digital Electronics, 4<sup>th</sup> Edition, Tata McGraw-Hill Education, 2010. [Unit 1, 2, 3, 4]
2. Donald P Leach, Albert Paul Malvino & Goutam Saha: Digital Principles and Applications, 7th Edition, Tata McGraw Hill, 2010.[Unit 5 & 6]

**Reference Books:**

1. Samir Palnitkar: Verilog HDL: A Guide to Digital Design and Synthesis, Second Edition, Prentice Hall PTR, 2003.
2. Stephen Brown, Zvonko Vranesic: Fundamentals of Digital Logic Design with VHDL, 2nd Edition, Tata McGraw Hill, 2005.
3. R D Sudhaker Samuel: Illustrative Approach to Logic Design, Sanguine-Pearson, 2010.
4. M Morris Mano: Digital Logic and Computer Design, 10th Edition, Pearson Education, 2008.





**SOLAPUR UNIVERSITY, SOLAPUR**  
**S. E. (COMPUTER SCIENCE & ENGINEERING)**  
**Semester – I**  
**CS 215 - COMPUTER GRAPHICS**

**Teaching Scheme**

Lectures: 3 Hrs/week, 3 credits  
Practical: 2Hrs/Week, 1 credit

**Examination Scheme**

ESE: 70 Marks  
ISE: 30 Marks  
ICA: 25 Marks

**Introduction:**

This course introduces the basic of computer graphics and different basic graphics functions. It also develops ability for implementation of different algorithms. With this course student can acquire computer graphics techniques, its uses and implementation details.

**Course Prerequisite:** Knowledge of C Programming & Mathematics.

**Course Objectives**

1. To understand basics of computer graphics.
2. To give more emphasis on implementation of Computer Graphics Algorithms.

**Course Outcome**

1. Student will able to know about computer graphics hardware and peripherals.
2. Student will able to implement 2D line and polygonal graphics and apply transformations.

**SECTION-I**

<b>UNIT - 1</b>	<b>Basic Concepts &amp; Devices</b>	<b>6 hrs</b>
	Introduction to computer graphics, Application of Computer Graphics, pixel, frame buffer, resolution, aspect ratio, Video display devices: Refresh CRT, Raster scan display, Random scan display, color CRT monitors, Interactive devices: joysticks, touch panels, light pens.	
<b>UNIT - 2</b>	<b>Raster scan Graphics</b>	<b>7 hrs</b>
	Line drawing algorithms: DDA, Bresenham's algorithm, Bresenham's Circle generation algorithm, Run Length Encoding, Polygon filling: Scan converting polygon, Edge fill, Edge flag, Seed fill.	
<b>UNIT - 3</b>	<b>Geometric Transformations</b>	<b>7 hrs</b>
	<b>2D Transformation:</b> Translation, Rotation, Reflection, Scaling, Shearing, Combined transformation, Rotation about an arbitrary point, Reflection through an arbitrary line.  <b>3D Transformation:</b> Scaling, Shearing, Rotation, Reflection, Translation, Multiple Transformation, Rotation about axis parallel to coordinate axis.	

## SECTION-II

UNIT - 4	Clipping & Display File Compilation	6 hrs
	Sutherland-Cohen line clipping algorithm, Midpoint subdivision algorithm, Viewing transformation, Window transformation, segmented display file, Display file compilation.	
UNIT - 5	Visible Lines & Visible Surfaces	6 hrs
	Hidden surfaces: introduction, back-face removal algorithm: Painter's algorithm, Warnock algorithm, Z-buffer. Antialiasing and antialiasing techniques, Half toning.	
UNIT - 6	Plane curves & Space curves	7 hrs
	Introduction to curve generation, Curve representation, interpolation, Non parametric & parametric curves, Bezier curves, B-spline curves, Introduction to fractals, Fractal lines and surfaces.	

### ICA:

Student should perform 8 to 10 experiments based on following guide lines.

1. To Study Basic graphics functions.
2. Implementation of DDA line drawing algorithm.
3. Implementation of Bresenham's line drawing algorithm.
4. Implementation of Bresenham's Circle generation algorithm.
5. Implement Polygon filling algorithms.
6. Implement 2D transformation.
7. Implementation of 3D transformation.
8. Implement Sutherland – Cohen line clipping algorithm.
9. Implementation of Warnock algorithm.
10. Case study of OpenGL
11. Implement a small animation package.

### Text Books:

1. Computer Graphics (**Chapter 1**) - Donald Hearn, Baker (second edition) PHI publications.
2. Procedural elements for Computer Graphics (**Chapter 2, 4, 5**) - David F. Rogers (second edition) Tata McGraw Hill publications.
3. Mathematical elements for Computer Graphics (**Chapter 3, 6**) - Rogers, Adams (second edition) McGraw Hill Publishing Company.

### Reference Books:

1. Computer Graphics with virtual reality systems - Rajesh K. Maurya.
2. Principals of Interactive Computer Graphics - William Newman, Sproull (second edition) McGraw-Hill Publication.



**SOLAPUR UNIVERSITY, SOLAPUR**  
**S. E. (COMPUTER SCIENCE & ENGINEERING)**  
**Semester – I**  
**CS 216 - ADVANCED C CONCEPTS**

**Teaching Scheme**

Lectures: 3 Hrs/week, 3 Credits  
Practical: 4 Hrs/Week, 2 Credits

**Examination Scheme**

POE: 50 Marks  
ICA: 25 Marks

**Introduction:**

This course introduces study and implementation of advanced features of C Programming language. Storage classes, built-in library in C, string processing, pointer concepts. This course also introduces few data structure concepts such as Recursion, file concepts and sorting, searching and hashing techniques.

**Course Prerequisite:**

Student shall have basic C programming knowledge. Student shall also have background of mathematical problem solving skill.

**Course Objectives:**

1. To study and use storage classes and macros.
2. To enable student understand recursive solutions for different types of problems.
3. To enable student understand and use file and pointer concepts.
4. To make student understand theory and practical aspects of searching and sorting algorithms.
5. To enable students to understand and implement hashing technique.

**Course Outcomes:**

1. Student can define and demonstrate storage classes in C.
2. Student can develop recursive solutions for given problems.
3. Student can implement file concepts and pointer concepts.
4. Student is able to describe and implement searching algorithms – linear, binary searching and hashing technique.
5. Student is able to describe and implement sorting algorithms –bubble sort, insertion sort, quick sort and merge sort.

**Section I**

<b>UNIT-1</b>	<b>C Preprocessor and Storage Classes</b>	<b>No of lectures – 05</b>
	Features of C Preprocessor, Macro Expansion - Macros with Arguments and Macros versus Functions, File Inclusion, Conditional Compilation - #if and #elif Directives; Storage Classes - Automatic, Register, Static, External storage classes.	

<b>UNIT-2</b>	<b>Pointers</b>	<b>No of lectures – 06</b>
	Define Pointer, Pointer to Pointer, Pointers to Array, Array of Pointers, Pointer to String, Pointer to Structure, Pointer to Function, Memory Allocation - Static and Dynamic memory – calloc(), malloc(), realloc() and free().	
<b>UNIT-3</b>	<b>Files</b>	<b>No of lectures – 04</b>
	Introduction, Streams and file types, File operations, Different File I/O Functions, other file functions, command line arguments.	
<b>UNIT - 4</b>	<b>Recursion</b>	<b>No of lectures – 05</b>
	Definition and Process of Recursion in C, Types of Recursion, Comparison of Iterative and Recursion methods, How recursion works, Factorial, Fibonacci sequence, Towers of Hanoi, Advantages and Disadvantages of recursive techniques.	

## Section II

<b>UNIT - 5</b>	<b>String Processing</b>	<b>No of lectures – 05</b>
	Declaration and initialization of strings, display of strings, string library functions in C – strlen(), strcat(), strcpy(), strcmp(); user-defined functions to perform string operations – length, compare, concatenate, copy; array of strings	
<b>UNIT - 6</b>	<b>Searching and Sorting</b>	<b>No of lectures – 07</b>
	Study and Implementation of Searching Algorithms- Linear search and Binary search. Study and Implementation of Sorting Algorithms- Bubble sort, Insertion sort, Merge sort, Quick sort, Selection sort, Shell sort and Radix sort.	
<b>UNIT - 7</b>	<b>Hashing</b>	<b>No of lectures – 07</b>
	Different Hash Functions, choosing a hash function Collision Resolution by Open Addressing: Linear probing, quadratic probing, double hashing, Collision Resolution by Chaining.	

### ICA:

ICA should be based on the following implementations using C programming language. Minimum 8 assignments should be conducted from following 11 assignments.

1. Assignments based on C Preprocessor and Macro concepts.
  - i) Write a C program to define macro definitions for the following:
    - (a) To test whether a character entered is a small case letter or not.
    - (b) To test whether a character entered is an upper case letter or not.
    - (c) To test whether a character is an alphabet or not. Make use of them macros you defined in (a) and (b) above.
    - (d) To obtain the bigger of two numbers.

- ii) Write a C program to define macro definitions with arguments for calculation of area and perimeter of a triangle, a square and a circle. Store these macro definitions in a file called “areaperi.h”. Include this file in your program, and call the macro definitions for calculating area and perimeter for different squares, triangles and circles.
2. Assignments based on Storage classes in C.
  - i) Write a C program to demonstrate storage classes such as auto, register, static and extern.
3. Assignments based on Pointer concepts.
  - i) Write a C program to demonstrate pointer concept.
  - ii) Write a C program to demonstrate pointer to pointer concept.
  - iii) Write a C program to demonstrate pointer to array concept.
  - iv) Write a C program to demonstrate array of pointers concept.
4. Assignments based on Pointer concepts.
  - i) Write a C program to demonstrate pointer to string concept.
  - ii) Write a C program to demonstrate pointer to function concept.
  - iii) Write a C program to demonstrate pointer to structure concept.
5. Assignments based on Dynamic memory allocation concepts in C.
  - i) Write a C program to demonstrate use of calloc() function.
  - ii) Write a C program to demonstrate use of malloc() function.
  - iii) Write a C program to demonstrate use of realloc() function.
  - iv) Write a C program to demonstrate use of free() function.
6. Assignments based on File concepts in C.
  - i) Write a C program to open a given file, read a file and display file contents.
  - ii) Write a C program to open a file and write to file.
7. Assignments based on Recursion concept. (any 2)
  - i) Write a C program to calculate factorial of given number using recursive method.
  - ii) Write a C program to display Fibonacci series using recursive method.
  - iii) Write a C program to solve Towers of Hanoi problem using recursion method.
  - iv) Write a C Program to find reverse of a number using recursion.
  - v) Write a C Program to convert decimal number to binary number using recursion.
  - vi) Write a C Program to find power of a number using recursion.
8. Assignments based on String operations.
  - i) Write a C program to perform different string operations using string library functions such as string length, comparison, concatenation, and copy
  - ii) Write a C program to perform different string operations such as string length, comparison, concatenation, and copy using user-defined functions.
9. Assignments based on searching concepts.
  - i) Write a C program to implement Linear Search.

- ii) Write a C program to implement Binary Search.
10. Assignments based on sorting concepts. (any 2)
- i) Write a C program to implement Bubble sort.
  - ii) Write a C program to implement Insertion sort.
  - iii) Write a C program to implement Quick sort.
  - iv) Write a C program to implement Merge sort.
  - v) Write a C program to implement Selection sort.
  - vi) Write a C program to implement Shell sort.
11. Assignments based on Hashing concepts using array (any 2)
- i) Write a C program to implement Hashing concept.
  - ii) Write a C program to demonstrate collision in hashing technique.
  - iii) Write a C program to demonstrate collision resolution using Open Addressing method i.e. Linear probing

**Text Books:**

1. Let Us C by Yashvant Kanetkar [Unit-1, 3, 5]
2. Pointers in C by Yashvant Kanetkar [Unit- 2]
3. Data Structure using C and C++ second edition by Yedidyah Langram , Moshe J, Augensteen, Aason M. Tanenbaum [Unit- 4, 7]
4. C with Data Structures by Dr. R. Nageswara Rao [Unit-1, 2, 3, 5, 6]

**Reference Books:**

1. Data Structures Using C & C++ by Rajesh Shukla.
2. Data Structures A Pseudocode Approach with C by Richard F. Gilberg & Behrouz A. Forouzan



**Solapur University, Solapur**  
**S.E. (Computer Sci. and Engg. / IT) Semester-II**  
**CS 221 APPLIED MATHEMATIC-II**

**Teaching Scheme**

**Lectures – 4 Hours/week, 4 Credits**

**Tutorial – 1 Hour/week, 1 Credit**

**Examination Scheme**

**ESE – 70 Marks**

**ISE – 30 Marks**

**ISA- 25 Marks**

---

**Introduction:**

This course introduces numerical methods (Unlike analytical methods) to solve algebraic and transcendental equations, simultaneous systems of linear equations and numerical integrations. This course also introduces the fuzzy set theory in brief which deals with characterizing the concept of uncertainty and its relationship to the increasingly important concept of information and complexity. This course also introduces simplex method to solve LPP and assignment problems.

---

**Course Prerequisite:**

Student shall have knowledge of basic notions of classical set theory and probability theory. Student shall have to be familiar with some analytical method for solving equations, simultaneous equations & analytical methods to solve definite integrations.

---

**Course Objectives:**

- 1) To make students familiar with the uncertainty.
- 2) To give students comprehensive coverage of operations on fuzzy sets.
- 3) To make student use of numerical methods for the problems that can not be solved analytically.
- 4) To enable students to solve Linear Programming Problems and Assignment Problems.

---

**Course Outcomes:**

- 1) Students can address the uncertainty.
- 2) Student can apply knowledge of basics of fuzzy set theory to solve the problems.
- 3) Student can demonstrate the fuzzification.
- 4) Student can implement knowledge of numerical methods to solve more complex problems in nature.
- 5) Students can evaluate a particular kind of problems arises in day to day life using simplex method and Assignment Problems.

---

**Section I**

**Unit 1: Solution of algebraic and transcendental equation**

No of lectures – 07

Basic properties of equations, False position method, Newton-Raphson method, Multiple roots, Newton's iterative formula for obtaining square root, system of non-linear equations by Newton Raphson method.

**Unit 2: Solution of linear simultaneous equations**

No of lectures – 07

Direct methods – Gauss Elimination method, Gauss Jordan methods, Method of factorization  
Iterative methods – Jacobi's method, Gauss-seidal method, power method to find eigen value and eigen vector.

**Unit 3: Numerical Integration**

No of lectures – 07

Numerical integration using Newton's cote's formulae – Trapezoidal rule, simpson's 1/3rd rule, Simpson's 3/8th rule, Weddel's rule, Romberg integrations, Double integrations.

**Section II****Unit 4: Classical (Crisp) sets to fuzzy sets:**

No of lectures – 07

Crisp sets, Basic types of fuzzy sets, Basic concepts of fuzzy sets, fuzzy sets vs Crisp sets:  
Additional properties of  $\alpha$  –cuts, representation of fuzzy sets and extension principle of fuzzy sets.

**Unit 5: Fuzzy arithmetic.**

No of lectures – 07

Fuzzy number, arithmetic operations on intervals, arithmetic operations on fuzzy numbers, fuzzy equations, lattice of fuzzy numbers.

**Unit 6: LPP and Assignment Problems.**

No of lectures – 07

Introduction of LPP, Simplex method for LPP, Assignment problem: introduction mathematical formulation of A.P., Hungarian method to solve AP.

**Internal Continuous Assessment (ICA):**

*ICA shall consist of minimum six to eight assignments based on entire curriculum*

**Text Books:**

- 1) B.S. Grewal, Numerical methods, Khanna publication, New Delhi.
- 2) George j Klir and Bo Yuan, Fuzzy sets and Fuzzy logic – PHI India.
- 3) Fundamental of statistics, S.C. Gupta, Himalaya house publication.

**Reference Books:**

- 1) George J. Klir and Tina A. Folger, Fuzzy sets, uncertainty and information, PHI India.
- 2) Robert J. Schilling, Sandra L. Harris, Applied Numerical methods for Engineers.
- 3) M.K. Jain, S.R.K. Iyengar, R.K. Jain, Numerical methods for scientific and engineering computations – New Age International ltd.
- 4) Pundir & Pundir, Fuzzy sets and their applications – Pragati Publications.





**Solapur University, Solapur**  
**S.E. (Computer Science & Engineering)**  
**Semester-II**

**CS 222 - THEORY OF COMPUTATION**

**Teaching Scheme**

Lectures– 3 Hours/week, 3 credits

Tutorial – 1 Hour/week, 1 credit

**Examination Scheme**

ESE – 70Marks

ISE – 30Marks

ICA - 25 Marks

---

**Introduction:**

Theory of computation lays a strong foundation for a lot of abstract areas of computer science. TOC teaches you about the elementary ways in which a computer can be made to think. Any algorithm can be expressed in the form of a finite state machine and can serve as a really helpful visual representation of the same. Sometimes, the finite state machines are easier to understand thus helping the cause furthermore

---

**Prerequisite:** Students should have prior knowledge of DMS

---

**Course Objectives:**

- 1) To describe the computational principles that are fundamentals to build finite machine.
- 2) To develop and analyze the construction of automata according to the computer science theory.
- 3) To outline the principles that will help students to develop best computer design

---

**Course Outcome:**

Upon completion of the course, students will be able to

- 1) Analyze and construct finite automata
- 2) Design and develop finite machine and find the output generated by the automata
- 3) Detect ambiguity in a system & overcome it.

---

**SECTION-I**

UNIT-1	Regular Expressions	5 hrs
	Regular expressions & corresponding regular languages, examples and applications, unions, intersection & complements of regular languages	
UNIT-2	Finite Automata	5 hrs
	Finite automata definition and representation, Non-deterministic F.A., NFA with $\lambda$ transitions, Equivalence of DFA & NFA	
UNIT-3	Kleen's Theorem	5 hrs
	Statements & proofs, minimizing number of states in an FA, Basics of Moore and Mealy Machines	
UNIT - 4	Grammars & Languages	5 hrs
	Definition and types of grammars and languages, derivation trees and ambiguity, CNF notations, Union, Concatenation and $*$ 's of CFLs, Eliminating $\lambda$ production and unit productions from a CFG, Eliminating useless variables from a Context Free Grammar.	

## SECTION-II

<b>UNIT – 5</b>	<b>Push down Automata</b>	<b>5 hrs</b>
	Definition, deterministic PDA & types of acceptance, equivalence of CFGs & PDAs.	
<b>UNIT – 6</b>	<b>CFL's &amp; Non CFL's</b>	<b>4 hrs</b>
	Pumping Lemma & examples, intersection and complements.	
<b>UNIT – 7</b>	<b>Turing machines</b>	<b>5 hrs</b>
	Models of computation, definition of TM as language Acceptors, Combining Turing machines, computing a function with a TM.	
<b>UNIT - 8</b>	<b>Variations in TM</b>	<b>5 hrs</b>
	TMs with doubly infinite tapes, more than one tape, Non-deterministic TM and universal TM.	

### ISE:

Students should solve assignments based on the topics below:

1. Regular Expression & Corresponding Languages
2. Union, Intersection & Complements of Regular languages
3. Design & Simulation of Simple Finite Automata
4. Nondeterministic Finite Automata & NFA with  $\wedge$  transitions ,Conversion of NFA to DFA
5. Draw NFA using Kleens theorem
6. DFA minimization
7. Grammer ,Removing ambiguity from a grammar ,Conversion to BNF & CNF form
8. Push Down Automata
9. Pumping Lemma & Examples for regular sets & regular languages
10. Turing Machine

### Internal Continuous Assessment (ICA):

*ICA shall consist of minimum ten assignments from the below list.*

1. Problems on obtaining a Regular Expression.
2. Problems on Constructing a FA (Deterministic Finite Automata, NFA and NFA- $\wedge$ 's).
3. Problems on Recursive definition of  $d^*$  for DFA, NFA and NFA- $\wedge$ .
4. Problems on Conversion of NFA to DFA, NFA- $\wedge$  to NFA and NFA- $\wedge$  to DFA.
5. Problems on Union, Intersection and Complements of a language i.e.  $L1 \cup L2, L1 \cap L2$  and  $L1 - L2$ .
6. Problems on Obtaining NFA and NFA- $\wedge$  using Kleene's Theorem Part – I for the given regular expression.
7. Problems on Obtaining a regular expression using Kleene's Theorem Part – II .
8. Problems on Minimization of Automata.
9. Problems on Derivation trees (leftmost derivation, rightmost derivation).
10. Problems on Construction of grammar (regular grammar and context free grammar).
11. Problems on nullable variables, useless symbols, unit productions.

12. Problems on conversion of given grammar into Chomsky Normal Form (CNF).
13. Problems on construction of PDA (by empty stack and by final state).
14. Problems on Pumping Lemma for Regular Languages and Pumping Lemma for Context Free Languages.
15. Problems on Construction of Turing Machine as acceptor and problems on Construction of Turing Machine as a computing function (ex.  $x + y$ ,  $x - y$ ,  $x * y$ , etc).

**Text Books:**

1. Introduction to languages & theory of computation -- John C. Martin (MGH)
2. Formal Languages & Automata Theory-- Basavraj S. Anami, Karibasappa K.G., Wiley Precise Textbook-Wiley India

**Reference Books:**

1. Theory of Computation—Rajesh K Shukla (CENGAGE Learning)
2. Introduction to Automata theory, languages and computations – John E. Hopcraft, Rajeev Motwani, Jeffrey D. Ullman (Pearson Edition).
3. Discrete mathematical structures with applications to Computer science -- J.P.Tremblay & R.Manohar (MGH)



**SOLAPUR UNIVERSITY, SOLAPUR**  
**S. E. (Computer Science & Engineering)**  
**Semester – II**  
**CS 223 - MICROPROCESSORS**

**Teaching Scheme**

Lectures: 3 Hrs/week, 3 Credits  
 Practical: 2Hrs/week, 1 Credit

**Examination Scheme**

ESE: 70 Marks  
 ISE: 30 Marks  
 ICA: 25 Marks

POE:50 Marks

**Introduction:**

This course introduces to develop an in-depth understanding of the operation of microprocessor , machine language programming and interfacing techniques.

**Prerequisite:** Knowledge of Digital Techniques and Basic Electrical Engineering

**Course Objectives**

1. To learn the evolution of microprocessor.
2. To understand the microprocessor architecture and its functionalities
3. To understand the programming model of 8086 microprocessor.
4. To develop the microprocessor based programs for various applications.
5. To make the interfacing in between microprocessor and various peripherals.
6. Understand the basics of other types microprocessors

**Course Outcome**

After the completion of the course the students will be able to

1. To understand the basic microprocessors architecture, its Functionality and interfacing with I/O devices
2. To understand the advanced processors and its Interfacing with various devices.
3. Get acquainted with computer organization and advanced systems.

**SECTION-I**

<b>UNIT-1</b>	<b>Introduction to Microprocessor</b>	<b>4hrs</b>
	Introduction to microprocessor, Features and pin diagram of 8085, 8085 MPU architecture,	
<b>UNIT-2</b>	<b>8086 Microprocessor</b>	<b>6 hrs</b>
	8086 Architecture, Internal Operation, Machine Language Instructions: Addressing Modes, Instruction Execution Timing	
<b>UNIT-3</b>	<b>Assembly Language Programming</b>	<b>6 hrs</b>
	Assembler Instruction Format,8086 instruction set, Directives	
<b>UNIT -4</b>	<b>System Bus Architecture</b>	<b>4 hrs</b>
	Introduction , Basic 8086 Configurations : Minimum Mode, Maximum Mode , System Bus Timing	

## SECTION-II

<b>UNIT-5</b>	<b>Interrupts</b>	<b>6 hrs</b>
	Introduction, Types of 8086 Interrupts, Interrupt and Interrupt service Routine, Maskable and Non-maskable Interrupt, Programmable Interrupt Controller (8259), Programmable, Features and Block Diagram of DMA Controller 8257, Operating modes of 8257.	
<b>UNIT-6</b>	<b>Programmable peripheral Interface</b>	<b>4 hrs</b>
	Programmable peripheral Interface 8255, Modes of Operation	
<b>UNIT-7</b>	<b>Programmable Communication Interface and 8087 NDP</b>	<b>4 hrs</b>
	8087 numeric data processor. NDP data types, processor architecture	
<b>UNIT-7</b>	<b>The 80286, 80386 and 80486 Microprocessor</b>	<b>5 hrs</b>
	Introduction to 80286 Microprocessor, Architecture, Real Address Mode Operation, Protected Mode operation, 80386 Architecture, 80486 Architecture	

### ICA:

Student should perform 8 to 10 experiments using TASM/MASM based on following guide lines.

1. Addition and subtraction of two 16 bit numbers
2. Addition and subtraction of two 32 bit numbers
3. 16 bit multiplication of unsigned numbers.
4. 8 bit division of unsigned numbers
5. Find factorial of number
6. Generate a Fibonacci series.
7. Program for block transfer
8. Program to arrange numbers in ascending and descending order
9. Program to find Largest No. in a block of data.
10. Program to display the string.
11. Program to implement key board sensing using 8255
12. Implementation of 7-segment display using 8255

### Text Books:

1. Microprocessor Architecture, Programming, and Applications with the 8085 - Ramesh Gaonkar, Fifth edition. (UNIT -1)
2. Lise & Gibson, "Microcomputer System 8086 / 8088" PHI, 2<sup>nd</sup> Edition. (Unit 2, 3, 4, 7)
3. D. V. Hall, "Microprocessor and Interfacing Programming & Hardware" TMH – 2<sup>nd</sup> Edition- (Unit 8)
4. A K Ray & K M Bhurchandi "Advanced Microprocessors and Peripherals". 2<sup>nd</sup> Edition (Unit 5,6)

### Reference Books:

1. Barry B. Brey, "The Intel Microprocessors: Architecture, Programming & Interfacing" PHI, 6th Edition.
2. Uffenback, "The 8086 Family Design" PHI, 2nd Edition.



# SOLAPUR UNIVERSITY, SOLAPUR

## S. E. (COMPUTER SCIENCE & ENGINEERING)

### Semester – II

## CS 224 - DATA STRUCTURES

### Teaching Scheme

Lectures: 3 Hrs/week, 3 Credits  
Practical: 4 Hrs/Week, 2 Credits

### Examination Scheme

ESE: 70 Marks  
ISE: 30 Marks  
ICA: 25 Marks  
POE: 50 Marks

### Introduction:

This course introduces various linear data structures like stack, queue, linked list and non-linear data structures like various kinds of trees and graphs. Course includes implementation of various operations of these data structures and some applications.

### Course prerequisites:

This course requires prior knowledge of advanced C.

### Course Objectives:

- 1) To enable students to understand various data structures.
- 2) To develop programming skills to implement and analyze linear and non linear data structures.
- 3) To strengthen the ability to identify and apply the suitable data structure for problem solving.

### Course Outcome:

- 1) Students will understand different data structures and difference between them.
- 2) Students will be able to implement different data structures by using various C language features.
- 3) Students will be capable to build real time applications using these data structures.

## SECTION I

UNIT - 1	Introduction to Data Structures & Stack	4 hrs
	What is Data Structure, types of data structures – static, dynamic, primitive, non-primitive, linear, non-linear Stack: Definition, representation, Operations, Implementation and applications like conversion of polish notations, evaluation of postfix expressions.	
UNIT - 2	Queues	4 hrs
	Definition, representation, Operations, Implementation: Linear Queue, Circular Queue, Priority Queue.	

<b>UNIT 3</b>	<b>Lists</b> Definition, representation, Operations, Types of Lists: Singly Linked list, Doubly Linked list, Circular Linked list, Stack using linked list, Queue using Linked list, Application of Linked list: Addition and Subtraction of two polynomials	
<b>UNIT - 4</b>	<b>Trees</b> Definition, Traversal, Linked implementation, Operations on: Binary trees and Binary Search Trees, Introduction to Threaded Binary trees.	<b>5 hrs</b>

## SECTION II

<b>UNIT - 5</b>	<b>Multiway Trees</b> Multiway search Trees, Balanced Multiway Trees, Traversing a Multiway Tree, Insertion in Multiway Tree: B Trees, B+ Trees	<b>7 hrs</b>
<b>UNIT - 6</b>	<b>Height Balance Trees</b> AVL Trees: Definition, Height of an AVL Tree, Insertion, Deletion of node in AVL Trees, Single and Double rotation of AVL Trees.	<b>7 hrs</b>
<b>UNIT - 7</b>	<b>Graphs</b> Definition, Undirected and Directed Graphs, Graph Terminologies, Computer Representation of Graphs, Graph Traversal methods: Depth First and Breadth First Search, Application: Shortest Path using Dijkstra's algorithm.	<b>6 hrs</b>

### ICA:

Student should perform minimum 18 assignments from the below list.

### List of Assignments:

1. Implementation of Stack using array
2. Implementation of Stack using structure.
3. Conversion of infix expression to postfix expression using Stack.
4. Evaluation of postfix expression using Stack.
5. Implementation of Linear Queue using array.
6. Implementation of Circular Queue using structures.
7. Implementation of Priority Queue.
8. Implementation of Singly Linked List and operations on it.
9. Implementation of Doubly Linked List and operations on it.
10. Implementation of Singly Circular Linked List.
11. Implementation of doubly Circular Linked List.
12. Implementation of Stack using Linked List.
13. Implementation of Queue using Linked List.
14. Implementation of polynomial addition using Linked List.

15. Implementation of Binary Search Tree and its traversal (In order, Preorder, Post order).
16. Implementation of Binary Search Tree and deletion of nodes in it.
17. Implementation of graph using adjacency matrix.
18. Implementation of graph using adjacency list.
19. Implementation of Depth First Traversal of Graph.
20. Implementation of Breadth First Traversal of Graph.
21. Creation of B Tree of order 3 or 5 by inserting nodes in it using some example.
22. Creation of B+ Tree of order 3 or 5 by inserting nodes in it using some example.
23. Creation of of AVL Tree by performing different rotations.

**Text Books:**

1. Data Structure and Program Design in C by Robert Kruse/C.L.Tonda/BruceLeung second edition, Pearson Education, Prentice Hall.
2. Data Structures: A Pseudo Approach with C. by Richard.F.Gilberg & Behrouz .A. Forouzan, second edition, Cengage Learning
3. Data Structure using C and C++ by Rajesh.K.Shukla, Wiley Publication

**Reference Books:**

- 1 Data Structures using C and C++, second edition by Yedidyah Langram, Moshe J, Augenstein, Aason .M. Tanenbaum.
- 2 Data Structures and Algorithms by Prof Maria .S. Rukadikar, Shroff Publications.
- 3 Data Structures Through C in Depth by S.K. Shrivastava, Depali Shrivastava, BPB Publications





**Solapur University, Solapur**  
**S.E. (Computer Science & Engineering)**  
**Semester-II**  
**CS 225 – COMPUTER NETWORKS**

**Teaching Scheme**

Lectures– 3 Hours/week, 3 Credits  
 Practical – 2 Hour/week, 1 Credits

**Examination Scheme**

ESE – 70Marks  
 ISE – 30Marks  
 ICA - 25 Marks

**Introduction:**

This course introduces TCP/IP protocol in detail and it also covers the IPv4 Addressing, Socket Programming, Transport layer and Application layer protocols.

**Course Prerequisite:** Student should have the knowledge of Data communication fundamentals and types of Computer Networks

**Course Objectives:**

1. To introduce the students with addressing in IP protocol.
2. To introduce the students with Transport layer protocols: TCP, UDP and SCTP.
3. To introduce the students with client-server paradigm for socket interfaces and to discuss the client-server communication using connectionless & connection-oriented services offered by the transport layer protocols.
4. To study the client server technology using installation and configuration of DHCP, DNS, FTP, TELNET and SSH.

**Course Outcomes:**

At the end of the course, student will be able to

1. To demonstrate the purpose of different layers.
2. To write application layer protocols using services offered by the transport layer protocols such as UDP, TCP & SCTP.
3. To show the functioning of DHCP, DNS and e- mail system

**SECTION-I**

<b>UNIT-1</b>	<b>IP Protocol</b>	<b>7 hrs</b>
	Internet Protocol: Introduction, IP Datagram, fragmentation, Addressing: Physical, Logical, Port & Application Specific Addresses. Introduction to IPv4 Addresses: Classful addressing, Classless addressing, Special addresses, NAT	
<b>UNIT-2</b>	<b>Transport Layer</b>	<b>7 hrs</b>
	<b>UDP:</b> Introduction, User Datagram, UDP Services, UDP Applications. <b>TCP:</b> TCP Services, TCP Features, Segment, A TCP Connection, State Transition Diagram, Window in TCP, Flow Control, Error Control, Congestion Control, TCP Timers. <b>SCTP:</b> Introduction, SCTP Services, SCTP Features, Packet Format	

<b>UNIT-3</b>	<b>Client Server Model and Socket Interface</b>	<b>6 hrs</b>
	<b>Client Server Paradigm:</b> Server, Client, Concurrency, Concurrency in Clients, Concurrency in Servers, Socket, Byte Ordering Functions. Socket System Calls, Connectionless Iterative Server, UDP Client Server Programs, and Connection-oriented Concurrent Server.	

## SECTION II

<b>UNIT- 4</b>	<b>Host Configuration &amp; Domain Name System</b>	<b>6 hrs</b>
	<b>Host Configuration:</b> BOOTP Operation, Packet format, DHCP: Introduction, DHCP Operation and Configuration. <b>Domain Name System:</b> Need for DNS, Name Space, DNS in the Internet, Resolution, DNS Messages, Types of Records.	
<b>UNIT- 5</b>	<b>Remote Login and TELNET</b>	<b>6 hrs</b>
	TELNET Concept, Time-Sharing Environment, Network Virtual Terminal, Embedding, Options, Symmetry, Sub option Negotiation, Controlling the Server, Out-of-Band Signalling, Escape Character, Mode of Operation, User Interface. <b>SSH:</b> Components, Port Forwarding, Format of SSH Packets.	
<b>UNIT- 6</b>	<b>File Transfer and Electronic Mail</b>	<b>7 hrs</b>
	<b>FTP:</b> Introduction, control & data connections, Communication over data and control connection, Command Processing <b>TFTP:</b> Messages, Connection, Data Transfer, UDP Ports, TFTP Applications. <b>Electronic Mail:</b> Architecture, User Agent, Message Transfer Agent, SMTP, Message Access Agent: POP and IMAP	

### ICA:

Students should perform minimum 8 experiments based on the following guidelines and preferably conducted on Unix / Linux platform using C language.

1. Installation of Unix/Linux Operating System.
2. Configuration of Network-Assigning IP Address, Subnet-Mask, Default Gateway, DNS Server Addresses & Testing Basic Connectivity.
3. Connectionless Iterative Server: C Implementation of Client-Server Programs Using Iterative UDP Server.
4. Connection-oriented Iterative Server: C Implementation of Client-Server Programs Using Iterative TCP Server.
5. Connection-oriented Concurrent Server: C Implementation of Client-Server Programs Using Concurrent TCP Server.
6. Implementation of Simple Network Chatting Application.
7. Remote Login: TELNET
  - a. Log on to a remote computer from client using TELNET.

b. After logging on executes few commands at remote server from client. For example user wants a server to display a file (hello.txt) on a remote server then he/she types: *cat hello.txt*.

c. Log on to a remote computer from client using TELNET and Putty terminal emulator. After logging on execute few commands. Here Client and Server are on heterogeneous systems, for example client is on windows and server is on Linux.

#### 8. Remote Login: SSH

a. Log on to a remote computer from client using SSH.

b. After logging on executes few commands at remote server from client. For example user wants a server to display a file (hello.txt) on a remote server then he/she types: *cat hello.txt*.

c. Log on to a remote computer from client using SSH and Putty terminal emulator. After logging on execute few commands. Here Client and Server are on heterogeneous systems for example client is on windows and server is on Linux.

#### 9. Installation and configuration of DHCP

#### 10. Installation and configuration of FTP.

### **Text Books:**

1. TCP/IP Protocol Suite: Behrouz A. Forouzan (Fourth Edition) (Unit 1,2,3,5,6)
2. TCP/IP Protocol Suite: Behrouz A. Forouzan (Third Edition) (Unit 4)
3. TCP/IP Protocol Suite: Behrouz A. Forouzan (Second Edition) (Unit 3)
4. Computer Networking: A Top-Down Approach Featuring the Internet, International Edition: James F. Kurose and Keith W. Ross

### **Reference Books:**

1. Internetworking with TCP/IP Vol III. Client-Server Programming & Applications: Douglas E. Comer
2. Data and Computer Communications: William Stallings
3. Data Communication and Networking: Behrouz A. Forouzan



**SOLAPUR UNIVERSITY, SOLAPUR**  
**S. E. (COMPUTER SCIENCE & ENGINEERING)**

**Semester – II**

**CS 226 - OBJECT ORIENTED PROGRAMMING THROUGH C++**

**Teaching Scheme**

**Theory : - 3 Hrs/Week, 3 credits**

**Practical : - 2 Hrs/Week, 1 credit**

**Examination Scheme**

**ICA :-25 Marks**

**POE: 50 Marks**

**Introduction:**

OOP provides a clear modular structure for program. OOP provides a good framework for code libraries where supplied software components can be easily adapted and modified by the programmer. C++ has a longer history with game development in general. It's a device level programming, Multi-paradigm, compiled, general-purpose programming language. It is regarded as a "middle-level" language, as it comprises a combination of both high-level and low-level language features.

**Course Prerequisite:**

Student should have programming language fundamentals.

**Course Objectives**

1. To learn fundamental concepts and principles of Object Oriented Programming.
2. To apply the OOP concepts for writing simple programs in C++.

**Course Outcome**

1. Students are able to comprehend and analyze simple C++ program.
2. Students are able to apply principle of OOP concept to develop complex C++ program.

**SECTION – I**

<b>UNIT - 1</b>	<b>OOP Concepts</b>	<b>5 hrs</b>
	Basic Concepts, Features, C++ Programming Basics, Object and Class, Array of objects, Constructors & types of constructors, Destructor.	
<b>UNIT - 2</b>	<b>Functions</b>	<b>5 hrs</b>
	Default & Reference arguments, Function Overloading, Inline functions, Return by reference, Friend Functions and Static Functions.	
<b>UNIT - 3</b>	<b>Operator Overloading</b>	<b>5 hrs</b>
	Definition, Overloading unary and binary operators, Overloading Extraction and Insertion Operators, Data Conversions.	
<b>UNIT - 4</b>	<b>Inheritance</b>	<b>5 hrs</b>
	Derived class and base class, Types of inheritance, Derived class constructors, Overriding Member Functions, Nesting of Classes.	

## SECTION – II

<b>UNIT – 5</b>	<b>Pointers and Virtual Functions</b>	<b>5 hrs</b>
	Memory Management Operators - new and delete, Pointers to objects, this pointer, Pointers to Derived classes, Virtual Function, Late Binding, Pure Virtual Functions, and Abstract Base Classes.	
<b>UNIT – 6</b>	<b>Manipulators and File Handling</b>	<b>5 hrs</b>
	C++ Streams, C++ stream classes, Managing Output with Manipulator, File Stream Classes, Working with File, Object I/O, Command Line Arguments.	
<b>UNIT – 7</b>	<b>Template and Exception Handling</b>	<b>5 hrs</b>
	Class Templates, Function template, Exception handling, Throwing Mechanism, Catching Mechanism, Rethrowing an Exception.	
<b>UNIT - 8</b>	<b>Introduction to Standard Template Library</b>	<b>4 hrs</b>
	Components of STL, Containers, Algorithms, Iterators, Application of Container classes.	

**Term Work:** Student should implement minimum 8 assignments from the following.

1. Assignment on Class, Constructor, Destructor
2. Assignment on function Overloading, Constructor Overloading
3. Assignment on Operator Overloading
4. Assignment on Multiple Inheritance, Multilevel Inheritance
5. Assignment on Static Variables & Function in class
6. Assignment on Virtual Function, Virtual Class
7. Assignment on Function template, Class template
8. Assignment on Friend class and function,
9. Assignment on File handling
10. Assignment on Exception handling.
11. An individual student has to submit a project using C++ language as a part of term work.

**Text Books:**

1. Object Oriented Programming with C++ - E. Balagurusamy (McGraw-Hill)
2. Programming with C++ - D. Ravichandran (TMGH)

**Reference Books:**

1. C++ programming language - Bjarne Stroustrup (AT & T)
2. Object oriented programming in Turbo C++ - Robert Lafore (Galgotia)