SOLAPUR UNIVERSITY, SOLAPUR

FACULTY OF ENGINEERING & TECHNOLOGY

INFORMATION TECHNOLOGY

Syllabus for

S.E. (Information Technology) w.e.f. Academic Year 2017-18

Choice Based Credit System

SOLAPUR UNIVERSITY, SOLAPUR
FACULTY OF ENGINEERING & TECHNOLOGY

Information Technology

Programme Educational Objectives and Outcomes

A. Program Educational Objectives

1. To build a strong foundation in mathematics, science & technology in students required to prepare them for Graduate studies and research.
2. To prepare students to apply knowledge of core & application domain, to analyze & design complex engineering problems using latest technologies & tools.
3. To develop effective communication, presentation skills and management principles in students and enable them to apply these in their work as a member or a leader in a team for managing projects.
4. To promote awareness for life-long learning, environment, sustainability, health & safety, economics etc. in students and to introduce them to professional ethics to build a good social personality.

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 Information Technology

Choice Based Credit System Syllabus Structure of S.E. Information Technology W.E.F. 2017-2018

Semester I

Course Code	***Theory Course Name***	***Hrs./week***			***Credits***	***Examination Scheme***			
		L	***T***	***P***		***ISE***	***ESE***	***ICA***	***Total***
IT211	Applied Mathematics - I	3	1	--	4	30	70	25	125
IT212	Discrete Mathematical Structure	3	1	--	4	30	70	25	125
IT213	Data Communication	3	--	--	3	30	70	--	100
IT214	Digital Logic Design	3	--	--	3	30	70	--	100
IT215	Computer Graphics	3	--	--	3	30	70	--	100
IT216	Programming in C	3	--	--	3	--	--	--	--
	Sub Total	18	2	--	20	150	350	50	550
ENV21	Environmental Science	1	-	-	-	-	-	-	1
Course Code	***Laboratory Course Name***								
							ESE		
							POE	***OE***	
IT213	Data Communication	--	--	2	1	--	50	--	25	75
IT214	Digital Logic Design	--	--	2	1	--	50	--	25	75
IT215	Computer Graphics	--	--	2	1	--		--	25	25
IT216	Programming in C	--	--	4	2	--	50	--	25	75
	Sub Total	--	--	10	5	--	150		100	250
	Grand Total	**18**	**2**	**10**	**25**	**150**	**500**		**150**	**800**

- *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:**

1. Student is required to study and pass Environmental Science subject in Second Year of Engineering to become eligible for award of degree.

2. Batch size for the practical /tutorial shall be of 20 students. On forming the batches, if the strength of remaining students exceeds 9, then a new batch shall be formed.

3. 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

4. Student shall select one Self Learning Module at T.E. Part I and T.E. Part II each from Technical and Humanities and Social Sciences Group with at least one Self Learning Module from the Humanities and Social Sciences Group

5. Curriculum for Humanities and Social Sciences Self Learning Modules is common for all under graduate programmes of faculty of Engineering and Technology

6. ICA 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

Faculty of Engineering & Technology

Second Year Information Technology

Choice Based Credit System Structure of S.E. Information Technology W.E.F. 2017-2018

Semester II

Course Code	***Theory Course Name***	***Hrs./week***			***Credits***	***Examination Scheme***			
		L	***T***	***P***		***ISE***	***ESE***	***ICA***	***Total***
IT221	Applied Mathematics – II	3	1	--	4	30	70	25	125
IT222	Theory of Computation	3	1	--	4	30	70	25	125
IT223	Microprocessor	3	--	--	3	30	70	--	100
IT224	Data Structures	3	--	--	3	30	70	--	100
IT225	Computer Networks	3	--	--	3	30	70	--	100
IT226	Object Oriented Programming through C++	3	--	--	3	--	--	--	--
ENV22	Environmental Science	1	-	-	-	-	-	-	1
	Sub Total	18	2	--	20	150	350	50	550
Course Code	***Laboratory Course Name***								
							ESE		
							POE	***OE***	
IT223	Microprocessor	--	--	2	1	--	50	--	25	75
IT224	Data Structures	--	--	4	2	--	50	--	25	75
IT225	Computer Networks	--	--	2	1	--	-	--	25	25
IT226	Object Oriented Programming through C++	--	--	2	1	--	50	--	25	75
	Sub Total		_	10	5		150		--	250
	Grand Total	**18**	**2**	**10**	**25**	**150**	**500**		--	**800**

- *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:**

1. Student is required to study and pass Environmental Science subject in Second Year of Engineering to become eligible for award of degree.

2. Batch size for the practical /tutorial shall be of 20 students. On forming the batches, if the strength of remaining students exceeds 9, then a new batch shall be formed.

3. 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

4. Student shall select one Self Learning Module at T.E. Part I and T.E. Part II each from Technical and Humanities and Social Sciences Group with at least one Self Learning Module from the Humanities and Social Sciences Group

5. Curriculum for Humanities and Social Sciences Self Learning Modules is common for all under graduate programmes of faculty of Engineering and Technology

6. ICA 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. (INFORMATION TECHNOLOGY)
Semester-I
IT 211- APPLIED MATHEMATICS-I

Teaching Scheme:
Lectures- 3 Hours / week, 3 Credits
Tutorial - 1 Hours / week, 1 Credit

Examination Scheme
ESE- 70 Marks
ISE - 30 Marks
ICA- 25 Marks

Introduction:

This course includes mathematical theory and concepts required by an Informational technology engineer. The course consists of linear differential equations which can be used for mathematical model which are appearing in information technology, 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 information technology 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:

Students will be able to**:**

1. Solve higher order linear differential equation with constant coefficient
2. Apply Laplace and inverse Laplace transforms for solving linear differential equations.
3. Express a function in terms of sine's and cosines components so as to model simple periodic functions.

4. Solve problems on Z transform and explain its properties
5. Find the relation between two variables for the given data using regression
6. Sketch and explain various probability distribution functions
7. Solve problems based on queuing theory

SECTION-I

UNIT-1	**Linear differential equations with constant coefficients:**	**7Hrs.**
	Basic definition, differential operator, complimentary functions, particular integral, Shortcut methods for standard functions like $e^{ax}, \sin(ax+b), \cos(ax+b), x^m, e^{ax}V \text{ and } xV$, particular integral by general method (without method of variation of parameters) for other functions.	
UNIT-2	**Z-Transform:**	**5 Hrs.**
	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**	**9 Hrs.**
	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:**	**7 Hrs.**
	Introduction, Definition, Euler's formula, Fourier series of periodic functions with period 2π and 2L, Dirichlet's theorem (only statement), even and odd functions, half range sine and cosine series.	
UNIT- 5	**Statistics and probability:**	**8 Hrs.**
	Coefficient of correlation and lines of regression of bivariate data, random variable, Binomial, Poisson, Normal distribution	
UNIT- 6	**Queuing Theory:**	**6 Hrs.**
	Polish expressions and their compilation, Groups, group codes.	

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. (INFORMATION TECHNOLOGY)
Semester-I
IT 212 - DISCRETE MATHEMATICAL STRUCTURES

Teaching Scheme	**Examination Scheme**
Lectures– 3Hours/week, 3 Credits	ESE – 70Marks
Tutorial – 1 Hour/week, 1 Credits	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:

7. To get acquainted to basic connectives and find equivalent formulas and normal forms.
8. To draw implications from basic primitives.
9. To introduce set theory and relations with illustrations.
10. To introduce the concepts of functions and its types through scenarios.
11. To define types of algebraic systems and applications.
12. To build algebraic structures like lattices, Boolean algebra and interpret them.

Course Outcomes:

Students will be able to**:**

1. Use connectives and develop well formed formulas and find the equivalence of formulas.
2. Convert given formula into its equivalent normal form.
3. Validate inferences for the given premises.
4. Apply set theory and relations to draw conclusions.
5. Define the function and apply it to different scenarios where functions can be used.
6. Demonstrate algebraic system and its properties, classifying them into semigroups, monoids and groups.
7. Generate group codes for 1 bit error detection and correction.
8. Identify and classify different types of Lattices and demonstrate Boolean algebra with its properties.

SECTION-I

UNIT-1	**Mathematical logic**	**6 Hrs.**
	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.	
UNIT-2	**Representation of expressions**	**4 Hrs.**
	Normal & Principle normal forms, completely parenthesized infix & polish notations, Theory of inference for statement calculus.	
UNIT-3	**Set theory**	**4 Hrs.**
	Basic concepts of set theory, types of operations on sets, ordered pairs, Cartesian product.	
UNIT - 4	**Relations**	**7 Hrs.**
	Relations, Properties of binary relations, Matrix and graph representation, Partition and covering of set, Equivalence relation, Composition, POSET and Hasse diagram.	

SECTION II

UNIT- 5	**Functions**	**4 Hrs.**
	Function -types, Composition of functions, Inverse functions.	
UNIT- 6	**Algebraic systems**	**7 Hrs.**
	Algebraic systems, semi groups and monoids, properties and example.	
UNIT- 7	**Groups**	**6 Hrs.**
	Polish expressions and their compilation, Groups, group codes.	
UNIT - 8	**Lattices and Boolean algebra**	**7 Hrs.**
	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.

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. (INFORMATION TECHNOLOGY)

Semester-I

IT 213 – DATA COMMUNICATION

Teaching Scheme	**Examination Scheme**
Lectures– 3 Hours/week, 3 Credits	ESE – 70Marks
Practical – 2 Hour/week, 1 Credits	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 introduce Data Communication Fundamentals such as Data, Signals, Transmission medias
2. To explain uses of Computer Network, OSI Reference model and TCP/IP model.
3. To demonstrate different physical media and devices.
4. To introduce different Data Link layer protocols.
5. To introduce different routing algorithms and congestion control in network Layer.

Course Outcomes:

Students will be able to**:**

1. Send data through various data communication modes.
2. Differentiate between the OSI reference model and TCP/IP model.
3. Identify and classify different physical media and devices.
4. Demonstrate functions of Data Link Layer.
5. Simulate different routing algorithms 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**	**8 Hrs.**
	DLL design issues, Error detection & correction, Elementary DLL protocols, Sliding window protocols.	

SECTION II

UNIT- 4	**Medium Access Control**	**8 Hrs.**
	Channel allocation problems, Multiple access protocol: ALOHA, CSMA, CSMA/CD, Collision free protocols, Limited contention protocols.	
UNIT- 5	**IEEE standards**	**6 Hrs.**
	IEEE standards 802.3, 802.4, 802.5 & 02.6(DQDB), Bridges, Switches, Routers	
UNIT- 6	**Network Layer**	**9 Hrs.**
	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	

Internal Continuous Assessment (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, subnetsmask, 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. (INFORMATION TECHNOLOGY)
Semester-I
IT 214 - DIGITAL LOGIC DESIGN

Teaching Scheme	**Examination Scheme**
Lectures– 3 Hours/week, 3 Credits	ESE – 70 Marks
Practical – 2 Hour/week, 1 Credit	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 in-depth knowledge of electronic components and fundamentals of digital electronics.

Course Objectives:

1. To introduce various number systems and Boolean algebra.
2. To introduce concepts 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 Outcomes:

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

UNIT-1	**Digital Principles and Number Systems**	**7 Hrs.**
	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.	
UNIT-2	**Combinational Logic Design**	**8 Hrs.**
	Introduction to Standard representation of Logical function, K-map representation, Simplification using k-map up to 4 variables, Minimization of logical function specified in minterms / 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.	
UNIT-3	**Combinational Logic Design using MSI Circuits**	**8 Hrs.**
	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

UNIT-4	**Sequential Logic Circuit**	**7 Hrs.**
	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.	
UNIT-5	**Introduction to Verilog HDL**	**8 Hrs.**
	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.	

UNIT-6	Digital Design with Verilog HDL	7 Hrs.
	HDL Implementation of Data Processing Circuits: 2:1, 4:1, 8:1 Multiplexer, BUS Representation in HDL, HDL Implementation of Flip-Flops, Register, Counters.	

Internal Continuous Assessment (ICA):

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 its 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, 4th 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, ZvonkoVranesic: 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. (INFORMATION TECHNOLOGY)
Semester – I
IT 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 introduce basics elements of computer graphics and graphic devices.
2. To demonstrate the line, circle and polygon filling algorithms.
3. To demonstrate 2D and 3D transformations.
4. To use clipping algorithms.
5. To introduce hidden and visible surfaces, different types of curves.

Course Outcome

Student will able to

1. Draw graphical elements using built-in graphic functions in ‘C’.
2. Differentiate different graphical devices.
3. Draw lines, Circles and fill polygons.
4. Apply simple 2D and 3D transformations to given object.
5. Demonstrate different clipping algorithms, surfaces and different types of curves.
6. Create simple 2D animations.

SECTION-I

UNIT-1	**Basic Concepts & Devices**	**6 Hrs.**
	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.	
UNIT-2	**Raster scan Graphics**	**8 Hrs.**
	Line drawing algorithms: DDA, Bresenham's algorithm, Bresenhams Circle generation algorithm, RunLengthEncoding, Polygon filling: Scan converting polygon, Edge fill, Edge flag, Seed fill.	
UNIT-3	**Geometric Transformations**	**9 Hrs.**
	2D Transformation: Translation, Rotation, Reflection, Scaling, Shearing, Combined transformation, Rotation about an arbitrary point, Reflection through an arbitrary line. **3D Transformation:** Scaling, Shearing, Rotation, Reflection, Translation, Multiple Transformation, Rotation about axis parallel to coordinate axis.	

SECTION-II

UNIT-4	**Clipping & Display File Compilation**	**7 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**	**7 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**	**7Hrs.**
	Introduction to curve generation, Curve representation, interpolation, Non parametric & parametric curves, Bezier curves, B-spline curves, Introduction to fractals, Fractal lines and surfaces.	

Internal continuous assessment (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. (INFORMATION TECHNOLOGY)
Semester – I
IT 216 – PROGRAMMING IN C

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 and background of mathematics.

Course Objectives:

1. To introduce storage classes and macros.
2. To apply recursion for different types of problems.
3. To enable student to use file and pointer concepts.
4. To study different searching and sorting algorithms.
5. To study different hashing techniques.

Course Outcomes:
Students will be able to

1. Define, use and compare storage classes in C.
2. Develop recursive solutions for given problems.
3. Apply file, pointer concepts and string functions for given problem statements.
4. Implement linear, binary searching and hashing techniques.
5. Implement different sorting algorithms like bubble sort, insertion sort, quick sort and merge sort.

SECTION I

UNIT-1	**C Preprocessor and Storage Classes**	**05 Hrs.**
	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.	
UNIT-2	**Pointers**	**06 Hrs.**
	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().	
UNIT-3	**Files**	**05 Hrs.**
	Introduction, Streams and file types, File operations, Different File I/O Functions, other file functions, command line arguments.	
UNIT - 4	**Recursion**	**05 Hrs.**
	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

UNIT - 5	**String Processing**	**05 Hrs.**
	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	
UNIT - 6	**Searching and Sorting**	**09 Hrs.**
	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.	
UNIT - 7	**Hashing**	**07 Hrs.**
	Different Hash Functions, choosing a hash function Collision Resolution by Open Addressing: Linear probing, quadratic probing, double hashing, Collision Resolution by Chaining.	

Internal Continuous Assessment (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 Shellsort.
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 YashvantKanetkar**[Unit-1, 3, 5]**
2. Pointers in C by YashvantKanetkar**[Unit- 2]**
3. Data Structure using C and C++ second edition by YedidyahLangram , 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. (INFORMATION TECHNOLOGY)
Semester-II
IT 221 APPLIED MATHEMATICS-II

Teaching Scheme
Lectures– 3 Hours/week, 3 Credits
Tutorial – 1 Hour/week, 1 Credit

Examination Scheme
ESE – 70Marks
ISE – 30Marks
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 introduce the concept of uncertainty.
2) To give students comprehensive coverage of operations on fuzzy sets.
3) To introduce to students numerical methods and their use for the problems that cannot be solved analytically.
4) To introduce methods to solve Linear Programming Problems and Assignment Problems.

Course Outcomes:
At the end of the course student will be able to

1) Use the concept of uncertainty to solve problems.
2) Apply knowledge of basics of fuzzy set theory to solve problems.
3) Illustrate fuzzification.
4) Use knowledge of numerical methods to solve more complex problems of the real world.
5) Evaluate a particular kind of problems arising in day to day life using simplex method and Assignment Problems.

SECTION-I

UNIT-1	**Solution of algebraic and transcendental equation**	**7Hrs.**
	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**	**7Hrs.**
	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**	**7Hrs.**
	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:**	**7 Hrs.**
	Crisp sets, Basic types of fuzzy sets, Basic concepts of fuzzy sets, fuzzy set vs Crisp sets: Additional properties of α –cuts, representation of fuzzy sets and extension principle of fuzzy sets.	
UNIT- 5	**Fuzzy arithmetic.**	**7 Hrs.**
	Fuzzy number, arithmetic operations on intervals, arithmetic operations on fuzzy numbers, fuzzy equations, lattice of fuzzy numbers.	
UNIT- 6	**LPP and Assignment Problems.**	**7 Hrs.**
	Introduction of LPP, Simplex method for LPP, Assignment problem: introduction mathematical formulation of A.P., Hungerian method to solve AP.	

Internal Continuous Assessment (ICA):

ICA consists of minimum six tutorials based upon above curriculum. Tutorial shall include numerical problems

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, uncertainity and information, PHI India.
2) Robert J. Schiling, 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. (INFORMATION TECHNOLOGY)

Semester-II

IT 222 - THEORY OF COMPUTATION

Teaching Scheme	**Examination Scheme**
Lectures– 3Hours/week, 3 credits	ESE – 70Marks
Tutorial – 1 Hour/week, 1 credit	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 Discrete Mathematical Structure

---

Course Objectives:

1. To introduce the computational principles to build regular expressions for given regular language.
2. To introduce different types of automata.
3. To explain regular and non-regular languages.
4. To introduce context free grammar.
5. To introduce different types of Pushdown automata and Turing machine.

Course Outcome:

Students will be able to

1. Build regular expression for a given language.
2. Design different types of automata.
3. Classify languages as regular and non regular language.
4. Detect ambiguity in a grammar and convert into unambiguous grammar and normal forms.
5. Design pushdown automata and Turing machine for a given language.

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**	**8 Hrs.**
	Finite automata definition and representation, Non-deterministic F.A., NFA with ^ 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**	**8 Hrs.**
	Definition and types of grammars and languages, derivation trees and ambiguity, CNF notations, Union, Concatenation and *'s of CFLs, Eliminating ^ production and unit productions from a CFG, Eliminating useless variables from a Context Free Grammar.	

SECTION-II

UNIT – 5	**Push down Automata**	**5 Hrs.**
	Definition, deterministic PDA & types of acceptance, equivalence of CFGs & PDAs.	
UNIT – 6	**CFL's & Non CFL's**	**4 Hrs.**
	Pumping Lemma & examples, intersection and complements.	
UNIT – 7	**Turing machines**	**5 Hrs.**
	Models of computation, definition of TM as language Acceptors, Combining Turing machines, computing a function with a TM.	
UNIT - 8	**Variations in TM**	**5 Hrs.**
	TMs with doubly infinite tapes, Multitape, Non-deterministic TM and universal TM.	

Internal Continuous Assessment (ICA) :

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 ^ transitions ,Conversion of NFA to DFA
5. Draw NFA using Kleenstheorm
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

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

References:

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. (INFORMATION TECHNOLOGY)

Semester – II

IT 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 introduce 8085 and 8086 microprocessor architectures and their functionalities.
2. To get acquainted to the programming model of 8086 microprocessor.
3. To develop microprocessor based programs for various applications.
4. To build the interfacing between microprocessor and various peripherals.
5. To introduce basics of 80286, 80386 and 80486 microprocessors.

Course Outcomes
Students will be able to

1. Demonstrate the basic microprocessor architecture and its functionality.
2. Write and implement simple assembly language programs.
3. Write and implement simple assembly language programs for interfacing.
4. Differentiate between 8086, 80286, 80386 and 80486 microprocessors.

SECTION-I

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

SECTION-II

UNIT-5	**Interrupts**	**6 Hrs.**
	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.	
UNIT-6	**Programmable peripheral Interface**	**6 Hrs.**
	Programmable peripheral Interface 8255, Modes of Operation	
UNIT-7	**Programmable Communication Interface and 8087 NDP**	**6 Hrs.**
	8087 numeric data processor. NDP data types, processor architecture	
UNIT-7	**The 80286, 80386 and 80486 Microprocessor**	**7 Hrs.**
	Introduction to 80286 Microprocessor, Architecture, Real Address Mode Operation, Protected Mode operation, 80386 Architecture, 80486 Architecture	

Internal Continuous Assessment (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. Lice & Gibson, "Microcomputer System 8086 / 8088" PHI, 2nd Edition. (Unit 2, 3, 4, 7)
3. D. V. Hall, "Microprocessor and Interfacing Programming & Hardware" TMH – 2nd Edition- (Unit 8)
4. A K Ray & K M Bhurchandi "Advanced Microprocessors and Peripherals". 2nd 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. (INFORMATION TECHNOLOGY)

Semester – II

IT 224 - DATA STRUCUTRES

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 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 introduce students to various data structures.
2. To develop programming skills to implement and analyze linear and non linear data structures.
3. To identify and apply the suitable data structure for problem solving.

Course Outcome:

Students will be able to

1. Demonstrate and differentiate between different data structures.
2. Demonstrate Programming skills to implement different data structures by using various C language features.
3. Identify specific data structure to solve given problem

SECTION I

UNIT - 1	**Introduction to Data Structures & Stack**	**5 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.	
UNIT - 3	**Lists**	**6 Hrs.**
	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	
UNIT - 4	**Trees**	**8 Hrs.**
	Definition, Traversal, Linked implementation, Operations on: Binary trees and Binary Search Trees, Introduction to Threaded Binary trees.	

SECTION II

UNIT - 5	**Multiway Trees**	**7 Hrs.**
	Multiway search Trees, Balanced Multiway Trees, Traversing a Multiway Tree, Insertion in Multiway Tree: B Trees, B+ Trees	
UNIT - 6	**Height Balance Trees**	**7 Hrs.**
	AVL Trees: Definition, Height of an AVL Tree, Insertion, Deletion of node in AVL Trees, Single and Double rotation of AVL Trees.	
UNIT - 7	**Graphs**	**8 Hrs.**
	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.	

Internal Continuous Assessment (ICA) :

List of Assignments: (minimum 18 assigments)

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 YedidyahLangram, Moshe J, Augensteen, 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, DepaliShrivastava, BPB Publications

Solapur University, Solapur

S.E. (INFORMATION TECHNOLOGY)

Semester-II

IT 225 – COMPUTER NETWORKS

Teaching Scheme	**Examination Scheme**
Lectures– 3 Hours/week, 3 Credits	ESE – 70Marks
Practical – 2 Hour/week, 1 Credits	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 in IPv4 addressing.
2. To introduce Transport layer protocols: TCP, UDP and SCTP.
3. To introduce client-server paradigm for socket interfaces
4. To introduce different application layer protocols like DNS, FTP and TELNET.

Course Outcomes:

Student will be able to

1. Identify different addressing modes using IPv4.
2. Implement client-server paradigm for socket interfaces using UDP, TCP & SCTP.
3. Use different functions of application layer protocols like DNS , FTP , Email and TELNET

SECTION-I

UNIT-1	**IP Protocol**	**7 Hrs.**
	Internet Protocol: Introduction, IP Datagram, fragmentation, Addressing: Physical, Logical, Port & Application Specific Addresses. Introduction to IPv4 Addresses: Classful addressing, Classless addressing, Special addresses, NAT	

UNIT-2	**Transport Layer**	**8 Hrs.**
	UDP: Introduction, User Datagram, UDP Services, UDP Applications. **TCP:** TCP Services, TCP Features, Segment, A TCP Connection, State Transition Diagram, Window in TCP, Flow Control, Error Control, Congestion Control, TCP Timers. **SCTP**: Introduction, SCTP Services, SCTP Features, Packet Format	
UNIT-3	**Client Server Model and Socket Interface**	**8 Hrs.**
	Client Server Paradigm: 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

UNIT- 4	**Host Configuration & Domain Name System**	**7 Hrs.**
	Host Configuration: BOOTP Operation, Packet format, DHCP: Introduction, DHCP Operation and Configuration. **Domain Name System**: Need for DNS, Name Space, DNS in the Internet, Resolution, DNS Messages, Types of Records.	
UNIT- 5	**Remote Login and TELNET**	**7 Hrs.**
	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. **SSH**: Components, Port Forwarding, Format of SSH Packets.	
UNIT- 6	**File Transfer and Electronic Mail**	**8 Hrs.**
	FTP: Introduction**,** control & data connections, Communication over data and control connection, Command Processing **TFTP:** Messages, Connection, Data Transfer, UDP Ports, TFTP Applications. **Electronic Mail:** Architecture, User Agent, Message Transfer Agent, SMTP, Message Access Agent: POP and IMAP	

Internal Continuous Assessment (ICA) :

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

1. Configuration of Network-Assigning IP Address, Subnet-Mask, Default Gateway, DNS Server Addresses & Testing Basic Connectivity.
2. Connectionless Iterative Server: C Implementation of Client-Server Programs Using Iterative UDP Server.

3. Connection-oriented Iterative Server: C Implementation of Client-Server Programs Using Iterative TCP Server.
4. Connection-oriented Concurrent Server: C Implementation of Client-Server Programs Using Concurrent TCP Server.
5. Implementation of Simple Network Chatting Application.
6. 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.
7. 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.
8. Installation and configuration of DHCP
9. 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. (INFORMATION TECHNOLOGY)

Semester – II

IT 226 - OBJECT ORIENTED PROGRAMMING THROGH C++

Teaching Scheme	**Examination Scheme**
Theory : 3 Hrs./Week, 3 credits	ICA : 25 Marks
Practical : 2 Hrs./Week, 1 credit	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 introduce fundamental concepts and principles of Object Oriented Programming.
2. To enable students to write simple programs in C++.
3. To introduce students where exactly C ++ (OOPS) should be used.

Course Outcome

Students will be able to

1. Illustrate principles of OOPS : Data Abstraction, Polymorphism, Inheritance and File Handling.
2. Implement OOPs Concepts through C++ programs.
3. Identify the applications of OOPS concepts for a given problem.

SECTION – I

UNIT - 1	**OOP Concepts**	**5 Hrs.**
	Basic Concepts, Features, C++ Programming Basics, Object and Class, Array of objects, Constructors & types of constructors, Destructor.	

UNIT - 2	**Functions**	**6 Hrs.**
	Default & Reference arguments, Function Overloading, Inline functions, Return by reference, Friend Functions and Static Functions.	
UNIT - 3	**Operator Overloading**	**6 Hrs.**
	Definition, Overloading unary and binary operators, Overloading Extraction and Insertion Operators, Data Conversions.	
UNIT - 4	**Inheritance**	**6 Hrs.**
	Derived class and base class, Types of inheritance, Derived class constructors, Overriding Member Functions, Nesting of Classes.	

SECTION – II

UNIT – 5	**Pointers and Virtual Functions**	**6 Hrs.**
	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.	
UNIT – 6	**Manipulators and File Handling**	**5 Hrs.**
	C++ Streams, C++ stream classes, Managing Output with Manipulator, File Stream Classes, Working with File, Object I/O, Command Line Arguments.	
UNIT – 7	**Template and Exception Handling**	**5 Hrs.**
	Class Templates, Function template, Exception handling, Throwing Mechanism, Catching Mechanism, Rethrowing an Exception.	
UNIT - 8	**Introduction to Standard Template Library**	**5 Hrs.**
	Components of STL, Containers, Algorithms, Iterators, Application of Container classes.	

Internal Continuous Assessment (ICA) :

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 ICA.

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)