DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

B.TECH- CSE

FOUR YEAR PROGRAM

CURRICULUM STRUCTURE & SYLLABI
(2023-2027 BATCH ONWARDS)

SCHOOL OF ENGINEERING AND TECHNOLOGY
MODY UNIVERSITY OF SCIENCE AND TECHNOLOGY
LAKSHMANGARH

B.Tech (Computer Science and Engineering)
Four Year Program
Academic Curriculum (2023-2027 Batch Onwards)

Curriculum Components

Component	Credits
1. University Core (4 Courses)	8
2. University Elective (2 Courses)	6
3. Basic Science (9 Courses)	24
4. Engineering Science (9 Courses)	21
5. Program Core (30 Courses)	69
6. Program Elective / MOOCs(5 Courses)	15
7. Project / Summer Internship Project (7 Courses)	17
8. Proficiency (Non-Graded)	6
Total	160

First Year - Autumn Semester

Course Code	Course Title	Contact Hours Per Week			Credits	ETE Duration (Hours)	Weightage			
		L	T	P			CW	MTE	PBL	ETE
UC 23.101	Human Values & Professional Ethics	2	0	0	2	3	10	40		50
MA 23.101	Mathematics-I	3	1	0	4	3	10	40		50
PY 23.101	Engineering Physics	3	0	0	3	3	10	40		50
EE 23.101	Basics of Electrical Engineering	3	1	0	4	3	10	40		50
CS 23.101	Programming for Problem Solving	3	0	0	3	3	10	40		50
PY 23.111	Physics laboratory	0	0	2	1	2	20	20	20	40
EE 23.111	Electrical Laboratory	0	0	2	1	2	20	20	20	40
CS 23.111	Programming Laboratory	0	0	4	2	2	20	20	20	40
ME 23.112	Mechanical Workshop	0	0	2	1	2	20	20	20	40
CY 23.102	Environmental Sciences	2	0	0	2	3	10	40		50
SF 101	Personal Grooming and Fine Dining(Non Grade)	0	0	2	1	2	50			50
	Total				**23**					

First Year -Spring Semester

Course Code	Course Title	Contact Hours Per Week			Credits	ETE Duration (Hours)	Weightage			
		L	T	P			CW	MTE	PBL	ETE
UC 23.102	Pragmatic English Communication	2	0	0	2	3	10	40		50
MA 23.102	Mathematics-II	3	1	0	4	3	10	40		50
CY 23.101	Engineering Chemistry	3	0	0	3	3	10	40		50
ME 23.101	Applied Mechanics	3	1	0	4	3	10	40		50
EC 23.101	Basics of Electronics Engineering	3	1	0	4	3	10	40		50
CS 23.102	Object Oriented Programming Through C++	3	0	0	3	3	10	40		50
ME 23.111	Computer Aided Engineering Drawing	0	0	4	2	2	20	20	20	40
CY 23.111	Chemistry Laboratory	0	0	2	1	2	20	20	20	40
EC 23.111	Electronics Laboratory	0	0	2	1	2	20	20	20	40
CS 23.112	Object Oriented Programming Laboratory	0	0	2	1	2	20	20	20	40
SF 102	Social Grooming, Home & Decor and Business Communication (Non Grade)	0	0	2	1	2	50			50
	Total				**25**					

Project Based Learning (PBL) in Group of 3-5 Students

Second Year - Autumn Semester										
Course Code	Course Title	Contact Hours Per Week			Credits	ETE Duration (Hours)	Weightage			
		L	T	P			CW	MTE	PBL	ETE
UC 23.201	Introduction to Management & Leadership	2	0	0	2	3	10	40		50
MA 23.201	Mathematics-III	3	1	0	4	3	10	40		50
CS 23.201	Discrete Mathematics	3	1	0	4	3	10	40		50
CS 23.203	Data Structures	3	0	0	3	3	10	40		50
CS23.205	General Purpose Programming	3	0	0	3	3	10	40		50
EC 23.202	Digital Electronics	3	0	0	3	3	10	40		50
CS 23.253	Data Structures Laboratory	0	0	2	1	2	20	20	20	40
CS 23.255	General Purpose Programming laboratory	0	0	2	1	2	20	20	20	40
EC 23.224	Digital Electronics Laboratory	0	0	2	1	2	20	20	20	40
SF 201	Personal Grooming Fine Dining and Home Safety (Non Grade)	0	0	2	1	2	50			50
					22					

Second Year -Spring Semester										
Course Code	Course Title	Contact Hours Per Week			Credits	ETE Duration (Hours)	Weightage			
		L	T	P			CW	MTE	PBL	ETE
UC 23.202	Entrepreneurship Development	2	0	0	2	2	10	40		50
UE	University Elective-I	3	0	0	3	3	20			80
CS23.204	Database Management Systems	3	0	0	3	3	10	40		50
CS 23.202	Advanced Programming	3	0	0	3	3	10	40		50
CS23.206	Theory of Computation	3	0	0	3	3	10	40		50
CS23.208	Computer Organization and Architecture	3	0	0	3	3	10	40		50
CS 23.211	Software Engineering	3	0	0	3	3	10	40		50
CS23.252	Advanced Programming Laboratory	0	0	2	1	2	20	20	20	40
CS23.254	Database Management Systems Laboratory	0	0	2	1	2	20	20	20	40
SF 202	Social Grooming and Professional Skills(Non Grade)	0	0	2	1	2	50			50
					22					

Project Based Learning (PBL) in Group of 3-5 Students

Third Year - Autumn Semester										
Course Code	Course Title	Contact Hours Per Week			Credits	ETE Duration (Hours)	Weightage			
		L	T	P			CW	MTE	ETE	
RM 23.101	Research Methodology	2	0	0	2	3	10	40	50	
CS23.301	Design and Analysis of Algorithms	3	1	0	4	3	10	40	50	
CS23.303	Operating System	3	0	0	3	3	10	40	50	
CS23.305	Cyber Security	3	0	0	3	3	10	40	50	
	Program Elective 1*	0	0	0	3		Based on NPTEL exam			
CS23.351	Design and Analysis of Algorithms Laboratory	0	0	2	1	2	20	40	40	
CS23.355	Cyber Security Laboratory	0	0	2	1	2	20	40	40	
CS23.353	Operating System Laboratory	0	0	2	1	2	20	40	40	
CS 23.373	Summer Internship-I**	0	0	0	2	-	100			
CS 23.371	Project-1	0	0	4	2	2	-	50	50	
SF 301	Civility and Ethics Business and Professional Grooming, Corporate Dinning Etiquettes (Non Grade)	0	0	2	1	2	50	0	50	
					22					

Third Year –Spring Semester									
Course Code	Course Title	Contact Hours Per Week			Credits	ETE Duration (Hours)	Weightage		
		L	T	P			CW	MTE	ETE
CS 23. 302	Artificial Intelligence and Machine Learning	3	0	0	3	3	10	40	50
CS23.304	Compiler Design	3	0	0	3	3	10	40	50
CS23.306	Computer Networks	3	0	0	3	3	10	40	50
CS 23.308	Digital Image Processing and Computer Graphics	3	0	0	3	3	10	40	50
CS 23.391	Blockchain Components and Architecture	3	0	0	3	3	10	40	50
CS 23.392	Colloquium	0	0	2	1	-	100		
CS 23.354	Compiler Design Laboratory	0	0	2	1	2	20	40	40
CS 23.356	Computer Networks Laboratory	0	0	2	1	2	20	40	40
CS 23.358	Digital Image Processing and Computer Graphics Laboratory	0	0	2	1	2	20	40	40
CS 23.372	Project-2	0	0	4	2	2	-	50	50
SF 302	Soft Skills and Interview Skills (Non Grade)	0	0	2	1	2	50	0	50
					21				

*The course may be taken through MOOC. NPTEL/SWAYAM/Coursera will be the authorized platform for this MOOC course. Minimum 8 weeks course is required for a 3 credit.After successful completion of these MOOC courses the students, shall, provide their successful completion NPTEL status/certificates to the University (COE) through HOD/NPTEL coordinator. The Course code of MOOC courses will be same as the course code given in NPTEL Certificate.

Fourth Year – Autumn Semester									
Course Code	Course Title	Contact Hours Per Week			Credits	ETE Duration (Hours)	Weightage		
		L	T	P			CW	MTE	ETE
UE	University Elective-II	3	0	0	3	3	20		80
	Program elective-2	3	0	0	3	3	10	40	50
	Program elective-3	3	0	0	3	3	10	40	50
CS 23.473	Summer Internship-II**	0	0	0	2	-	100		
CS 23.471	Capstone Project-1	0	0	4	2		100		
					13				

Fourth Year –Spring Semester									
Course Code	Course Title	Contact Hours Per Week			Credits	ETE Duration (Hours)	Weightage		
		L	T	P			CW	MTE	ETE
	Program elective-4	3	0	0	3	3	10	40	50
	Program elective-5	3	0	0	3	3	10	40	50
CS23.472	Capstone Project-2	0	0	12	6			50	50
	Total				**12**				

** Summer Internship-I and II must be performed in an Industry between summer vacation of end of Second Year and end of Third Year.

Credit Distribution across all Components								
Semester	UC	UE	BS	ES	PC	PE	Proj/SIP	Total
First	2		10	6	5			23
Second	2		8	11	4			25
Third	2		4	4	12			22
Fourth	2	3			17			22
Fifth			2		13	3	4	22
Sixth					18		3	21
Seventh		3				6	4	13
Eighth						6	6	12
Total	8	6	24	21	69	15	17	160

UC	University Core
UE	University Elective
BS	Basic Science
ES	Engineering Science
PC	Program Core
PE	Program Elective / MOOC
Proj/SIP	Project/Summer Internship

Elective 1	
	MOOC

Elective 2	
CS 23.412	Big Data and Unstructured Databases
CS 23.413	Distributed Databases
CS 23.462	Mobile App Development (Infosys)

Elective 3	
CS 23.431	Advanced Web Technologies
CS 23.441	Shell Programming
CS 23.461	Cloud Computing (Infosys)

Elective 4	
CS 23.493	Foundations and Applications of IoT
CS 23.495	Principles and Practices of Secure Computing
CS 23.463	Agile Software Development (Infosys)

Elective 5	
CS 23.494	Software Defined Networking
CS 23.464	Business Intelligence and its application (Infosys)
CS 23.465	Machine Learning using R (Infosys)

CS 23.101 **Programming for Problem Solving** **Total Lectures: 40**

Credits: 3-1-0-4

Prerequisite(s): NIL

Objective(s):

1. To learn principles of programming by the structured native language 'C'.
2. To comprehend the rudiments, constructs, syntax, and statements with essentials of efficient compiling, running and debugging processes.

1. **Introduction:** The role of programming languages, programming paradigms, language description and syntax structure, basics of structured and object-oriented programming, functional programming, logic programming, concurrent programming. [5]
2. **Fundamentals of 'C' language**: Character set, variables data types constants, arithmetic, relational and logical operators, type conversion,increment,decrement, bitwise and assignment operators, conditional expressions, statements and blocks, *if-else*, *switch*, loops-*do-while*, *while*, *for*; *break*, *continue* and *goto* statements, labels. [10]
3. **Functions, pointers and Arrays**: Basics of functions, storage classes, scope rules, and header files, recursion and C preprocessor, arrays, pointers and addresses, types of pointers, pointers and arrays address arithmetic, character pointers and functions, pointer arrays, pointers to functions, function returning pointer, pointer to array, array of pointers, and multidimensional arrays, command line arguments, dynamic memory allocation-malloc(), calloc(), realloc(), free() [13]
4. **Structures and unions:** Basics of structures and functions, arrays of structures, pointers to structures, passing structure to function, function returning structure, union, self-referential structures. [6]
5. **I/O and File Processing:** Concept of files, standard input and output, formatted output, file access in different modes, reading from and writing to a file, random access file operations. [6]

Outcome(s): Students would be able to apply learning and knowledge of C language analytically, critically and logically to solve mathematical, scientific and real-life problems in an efficient manner.

Text Books:

1. Brain W. Kernighan and Dennis M. Ritchie, The C Programming Language, Second edition, Pearson Education.
2. Ravi Sethi & K.V.Viswanatha, Programming Languages-Concepts, and Constructs, Second edition, Pearson Education,

Reference Books:

1. Herbert Schildt, C: The Complete Reference, Fourth Edition, McGraw-Hill Education,
2. E Balagurusamy, Programming in ANSI C, Tata McGraw - Hill Education,

CS 23.102 **Object Oriented Programming Through C++** **Total Lectures: 40**

Credit: 3-0-0-3

Prerequisite(s): Programming for problem Solving-I

Objective(s):

1. To familiarize the students with basic concepts of object oriented Programming and developer tools.
2. To present the syntax and semantics of the C++ Programming.

1. **Object Oriented Programming:** Basic concepts of object–oriented Programming, Structured versus object-oriented programming, merits and demerits of object oriented methodology. [6]
2. **Beginning With C++:** Preprocessor directives, variables, data types, constants, operators, scope resolution operator, memory management operators, decision and loop controls, arrays, strings, function, inline function. [6]
3. **Classes And Objects:** Class definition, structure versus class, access specifiers, accessing member functions within a class, data hiding, arrays within a class, static data and member functions, objects, friend functions and friend classes, constant member functions, pointers to members, constructors and destructor. [8]
4. **Overloading and Exception Handling:** Definition, unary operator overloading, binary operators overloading, overloading of new and delete operators, manipulation of string using operators, type conversion, overloading with friend functions, function overloading, error and exception handling. [8]
5. **Inheritance and Overriding:** Derived and base classes, type of inheritance, derived class constructors and destructor, overriding member functions, visibility modes public protected and private inheritance, virtual base classes, abstract classes, this pointer, virtual functions and pure virtual functions, static and dynamic bindings, **I/O Streams and The Standard Library:** Header files, I/O stream library, stream I/O for objects, File I/O [12]

Outcome(s): At theendofcourse,studentshaveagoodunderstandingabouttheconceptofobject-oriented programming using C++and able to write and read C++ code for real world applications

Text Books:

1.HerbertSchildt,“C++:The Complete Reference”, TMH
2.Robert Lafore, “Object-Oriented Programming in C++”, SAMS

Reference Books:

1.Bjarne Stroustrup**, ”**Programming-Principles and PracticeUsing C++”,Addison- Wesley.
2.E.Balagurusamy,“ObjectOriented Programming With C++”,TMH

CS 23.111 **Programming Laboratory** **Total Lectures: 26**

Credit:0-0-4-2

Objective:
- To make students familiar with syntax and structure of C-programming.
- To learn problem solving techniques using C programs implementation.

Prerequisite: Nil

1. Implement the logic of Data Types & Operators.
2. Implement the logic of Type Conversions.
3. Implement the logic of Arithmetic, Relational, and logical operators.
4. Implementation of Bitwise, unary, assignment operators.
5. Implementation of Increment and Decrement operators.
6. Implementation of Conditional operators.
7. Implement the logic of Increment and Decrement operators.
8. Implement programs in order to cover Control Structure, If-Else statements, Nested If-Else statements, Switch-case statement, Loops- For, While, Do-While, Break Continue, Go-To Statements, and Labels.
9. Implement programs in order to cover processing an array, passing arrays to functions, Multidimensional arrays.
10. Implement programs to cover all possible String operations.
11. Implement programs in order to cover Function prototypes and passing arguments to a function, Recursion.
12. Implement programs in order to cover Pointers declaration, operations on pointers, Array of pointers, pointers to arrays.
13. Implementation of structure and union programs.
14. Implement the use of Storage classes in C programming.
15. Implement programs for Reading and writing from/to a file.
    Implement random access file operation

Outcome: On completion of this course student will be able to:
- Apply learning and knowledge of C language analytically, critically and logically to solve mathematical, scientific and real-life problems in an efficient manner

CS23.112 **Object Oriented Programming Laboratory** **Credit: 0-0-2-1**

Prerequisite CS 23.111 (Programming Laboratory-I)

Objective: The objective of this lab is to provide students with an understanding of the fundamental concepts in object oriented programming and to provide the skills needed for developing the software application.

Contents: Lab manual based upon the theory course **CS23.102** and programs based on it (class and objects, overloading, overriding, friend class and function, inheritance, exception handling, template, I/O File etc.) to achieve the listed course objectives and outcomes.

1. Understand the basics of C++ library, variables, data input-output.
2. Implement class and object concepts
3. Implement the concept of polymorphism using function overloading.
4. Implement code to understand friend function and friend class
5. Program to understand different types of constructors and destructor.
6. Programs to understand the usage of constant data member and member function, static data member and member function in a class.
7. Implement different types of inheritance
8. Implement overriding and virtual function.
9. Implement Operator overloading concepts.
10. Write programs to understand function template and class template.
11. Write programs to understand exception handling techniques.
12. Write programs to understand file handling techniques.

Outcomes: Upon completion of the course students will have knowledge of the underlying the static member, friend function, operator overloading, exception handling, inheritance and polymorphism in object oriented programming application.

CY 23.101 **Engineering Chemistry** **Total Lectures: 39**

Credit: 3-0-0-3

Prerequisite(s): NIL

Objective(s):

- To purify the most valuable natural resource of the future-water.
- To understand one of the most versatile Engineering Materials: Polymers their preparation, properties and uses.
- To develop the basics of protecting engineering materials from the natural forces.
- To understand the present day up-to-date knowledge of the contemporary energy sources.
- To build up the basis of the future Engineering materials like superconductors, conducting organic polymers, Nano materials etc.

1. **Water treatment:** Hardness, types of hardness, degree of hardness and its estimation (EDTA methods). Numerical problems on degree of hardness. Boiler troubles: sludge and scale formation, Prevention of sludge and scale formation in boilers. Softening of hard water (Lime-Soda, Permutit, Ion Exchange and calgon methods). Numerical problems based on water treatment (Lime-Soda process). [8]

2. **Polymers and Phase Rule:** Industrial applications of Inorganic and organic polymers. Phase Rule and its application to water and Sulfur system. [8]

3. **Electrochemistry and Corrosion:** Electrochemical Changes, Electrochemical cell and its classifications (Galvanic cell, electrolytic cell). Types of electrodes. Origin of electrode potential, Electrochemical Series. Electromotive force, Nernst equation of electrode potential. [8]
   Corrosion: Definition and its significance, Theories of corrosion, Mechanism of corrosion, types of corrosion, Corrosion control (Corrosion inhibitors, Cathodic protection - sacrificial anodic and impressed current cathodic protection).

4. **Fuels and Lubricants:** Introduction - Importance of fuels in Industries. Classifications of fuels, Comparison of solid, liquid & solid fuels, Calorific value of a fuel. Determination of calorific value by Bomb calorimeter. Refining and cracking of petroleum, knocking. Octane Number and Cetane Number. L.P.G., Coal gas and Producer gas. [9]
   Lubricants: Introduction and Classification of Lubricants. Properties of Lubricants, Lubricant Oil, Grease, Emulsions.

5. **New Engineering Materials:** Introduction: Superconductors, high and low temperature superconductors, organic electronic materials, Fullerenes and Nano materials, Industrial applications of fullerenes and Nano materials. [6]

Outcome(s): The student will be able to understand the basis of the future Engineering materials like of lubricants, superconductors, conducting polymers, Nano materials etc and purification of the most valuable natural resource of the future-water.

Text Books:

1. Jain and Jain, "Engineering Chemistry," 16th Edition, Dhanpat Rai Publishing Company, 2015.
2. S.S. Dara, "A Text Book of Engineering Chemistry," 12th Edition, S. Chand & Company Limited, 2014.

Reference Books:

1. Peter Atkins, Julio de Paula, and James Keeler, "Atkins' Physical Chemistry" Eleventh Edition, Oxford University Press, 2017.
2. William D. Callister, "Materials Science and Engineering: An Introduction, Eighth Edition, John Wiley & Sons, Inc., 2010.

CY 23.102 **Environmental Sciences** **Total Lectures: 26**

Credit: 2-0-0-2

Objective:
- To teach the principal resources, renewable energy systems and biosphere.
- To explore the environmental impact of various energy sources and also the effects of different types of pollutants.

Pre-requisite: Nil

1. **Basic concepts of Environmental Studies:** Definition, scope and importance**,** Multidisciplinary nature of environmental studies, Need for public awareness. Role of an individual in conservation of natural resources. Equitable use of resources for sustainable lifestyles. [4]
2. **Natural Resources:** Renewable and non-renewable resources. **Mineral resources:** Use and exploitation, environmental effects of extracting and using mineral resources, case studies. **Forest resources:** Use and over exploitation, deforestation, case studies. **Land resources:** Land as a resource, land degradation, man induced landslides, soil erosion and desertification. **Food resources:** World food problems, changes caused by agriculture and overgrazing, effects of modern agriculture, fertilizer-pesticide problems, [6]
3. **Ecosystems & Biodiversity:** Concept and Structure and function of an ecosystem. Producers, consumers and decomposers. Energy flow and nutrient cycling in the ecosystem. Food chains, food webs and ecological pyramids.
   Biodiversity and its conservation: Introduction – Definition, genetic, species and ecosystem diversity. Threats to biodiversity: habitat loss, illicit poaching, man-wildlife conflicts. [5]
4. **Environmental Pollution :** Definition, cause, effects and control measures of: Air pollution, Water pollution, Soil pollution, Noise pollution, Thermal pollution, Nuclear Radiation.
   Solid waste Management: Causes, effects and control measures of urban and industrial wastes. Pollution case studies. Disaster management: floods, earthquake, cyclone and landslides. [6]
5. **Social Issues, Population and impacts of Environmental impacts:**Urban problems related to energy, Resettlement and rehabilitation of people, Global Climate Change, global warming, acid rain, ozone layer depletion. Wasteland reclamation. Population growth, variation among nations. Population explosion, Environment and human health. Human Rights. Value Education. Women and Child Welfare. Role of Information Technology in Environment and human health. Case Studies. Environment Protection Acts. [5]

Outcome: At the end of this course, students will demonstrate the ability to
- Evaluate local, regional and global environmental topics related to resource use and management
- Propose solutions to environmental problems related to resource use and management.
- Describe threats to global biodiversity, their implications and potential solutions.

Text Books:

1. Erach Bharucha “Environmental Studies”, University Grants Commission, 2004.
2. Anubha Kaushik, C P Kaushik “ Perspectives In Environmental Studies” New Age International Publishers. 2018

Reference Books:

1. C. William, P. Saigo, B Woodworth, “Environmental Science: A Global Concern”, Sixth Edition McGraw-Hill Higher Education, 2001.
2. Botkin, and Keller, “Environmental Science” Wily Publication, 2012.
3. R.T. Wright, D.F. Boorse, “Environmental Science” 11th Edition, PHI, 2010.

CY 23.111 **Chemistry Laboratory** **Total Hours:26**

Credit: 0-0-2 - 1

Prerequisite(s): NIL

Objective(s):

- To introduce students to water chemistry and purify the most valuable natural resource of the future-water.
- To understand one of the most versatile Engineering Materials: Polymers, lubricants, fuel and their properties and uses.

1. To determine the percentage of Available Chlorine in the given sample of Bleaching powder. Iodometrically.
2. To determine the alkalinity of water sample.
3. To prepare urea formaldehyde resin.
4. To determine the strength of given HCl solution by titrating it against NaOH solution using pH meter.
5. To determine the strength of given acid solution by titration it against sodium hydroxide solution using conductometer.
6. To determine molecular weight of a polymer using Ostwald viscometer.
7. Determination of viscosity of given lubricating oil by Redwood viscometer.
8. Determination of cloud and pour point of given lubricating oil.
9. Determination of the flash and fire point of given lubricating oil.
10. Determination of phase diagram of water system.
11. To determine the calorific value (HCV) of a sample of fuel.
12. To determine the temporary and permanent hardness in the given water sample by complexometric titration using EDTA as standard solution.

Outcomes The students will learn to:

- Estimate the Hardness of water, alkalinity of water.
- Familiar with the important practical applications of electrochemistry, lubricants, solids, their properties and applications, and the polymer materials.

EC 23.101 **Basics of Electronics Engineering** **Total Lectures: 40**

Credit:3-1-0-4

Prerequisite Basic Knowledge of Engineering and Applied Physics

Objective: To learn the fundamental principle and applications of electronics and communication engineering and devices.

1. **Introduction:** Brief history of major inventions in ECE, Electronic test and measurement equipment's, Specifications for electronic components. [3]
2. **Semiconductors**: Material and its properties, Formation of energy band diagram, Currents in semiconductors, Fermi-Dirac energy distribution, Types of semiconductors – Intrinsic and Extrinsic, Mass action law, Hall effect. [7]
3. **Diodes**: Introduction, Characteristic of ideal and real diode, Current components in diode, load characteristics, Breakdown in diode, Zener diode, Varactor diode, Tunnel diode, Clippers and Clampers, Rectifiers – Half and Full-wave rectifiers. [10]
4. **Transistors:** Introduction to pnp and npn transistors, current components, active, cutoff and saturation regions, CC, CB and CE configuration, Operating point, Biasing circuits, Bias Stability, Thermal runway and thermal stability. [10]
5. **Introduction to Signals and Communication Systems**: Definition, Classification of signals – Analog and Digital, Deterministic and Random signals, Power and Energy of signals, Bandwidth, Bit and Baud rate. Block representation of communication system, Modulation – Definition, Need of modulation, Types of modulation. [10]

Outcome: After completion of this course, students would be able to :

1. Distinguish between conductors, semiconductors and insulators based on energy band theory and classify different types of semiconductors.
2. Employ the concept of Energy Band Theory and Fermi Levels to explain the operating principle of semiconductors.
3. Demonstrate the operating principle and characteristics of p-n junction diodes, different diode circuits and Transistors.
4. Understand various types of communication signals and its characteristics parameters.
5. Understand the basic concepts of communication system and modulation techniques..

Text Books:

1. George Kennedy, Bernard Davis, "Electronics Communication Systems", Tata McGraw-Hill, 3rd Edition, 2007.
2. Behrouz Forouzan, "Data Communication and Networking", McGraw-Hill, 4th Edition, 2006.

Reference Books:

1. David A. Bell, "Electronics Devices and Circuits", PHI, 4th edition, 2008.
2. Wayne, Tomasi, "Advanced Electronic Communication Systems", PHI, 6th edition, 2013.
3. Robert L. Boylestad, Louis Nashelsky, "Electronics Devices and Circuits Theory", Prentice Hall, 10th Edition, 2009.

EC 23.111 **Electronics Laboratory** **Total Hours: 26**

Credit: 0-0-2-1

Objective:

- To gain practical knowledge by applying the experimental methods to correlate with the theory of Electronics Engineering.
- To apply the analytical techniques and graphical analysis to the experimental data.
- To enable the students to simulate and test the Electronics circuits using Multisim software.

Prerequisite: Basic Understanding of Engineering Physics and Electronics

Practical to be performed on Hardware:

1. Familiarity of Soldering/de-soldering process, Component identification, testing and their verification using multimeter.
2. To plot V-I characteristics of junction diode and Zener diode.
3. To design a half wave rectifier with and without filter circuit and find the ripple factor.
4. To design a full wave rectifier with and without filter circuit and find the ripple factor.
5. Study of clipper and clamper circuits and their implementation on breadboard.
6. To plot the characteristics of Transistor under Common Emitter configuration

Practical to be performed using Multi-Sim Software:

7. To verify Ohm's law and Kirchhoff's law
8. To Simulate half wave and full wave bridge rectifier.
9. To plot V-I characteristics of p-n junction diode.
10. To plot V-I characteristics of Zener diode.
11. Study of Clipper circuit and its implementation on Multisim.

Outcome: After performing the above experiments, Student will be able to :

1. Identify and describe basic electronic components used in circuits.
2. Build and configure basic electronic circuits using breadboards.
3. Analyze the voltage-current characteristics of p-n junction diodes , Zener diodes and Transistor.
4. Develop clipper and clamper circuits to achieve desired voltage waveforms.
5. Utilize Multisim software to simulate and validate circuit designs.

EE 23.101 **Basics of Electrical Engineering** **Total Lectures: 40**

Credit: 3-1-0-4

Objective: This course facilitates the students to get a comprehensive exposure to various concepts and devices of electrical engineering.

Pre-requisite: Nil

1. **DC Circuits:** Electrical circuit elements (R, L and C), voltage and current sources, Ohm's law, Kirchhoff current and voltage laws, analysis of simple circuits with dc excitation using mesh and nodal method. Network Theorems - Superposition, Thevenin, Norton and Maximum power transfer Theorems. Time-domain analysis of first-order RL and RC circuits. Star-Delta Conversion [8]
2. **AC Circuits:** Basic terminologies, representation of sinusoidal waveforms, average, rms and peak values of a.c. quantities, phasor representation, real power, reactive power, apparent power, power factor. Analysis of single-phase ac circuits consisting of R, L, C, RL, RC, RLC combinations (series and parallel). [8]
3. **Three Phase A.C. Circuits:** Three phase supply, phase sequence, star connection, delta connection, relationship between phase and line quantities in 3-phase circuits, measurements of three phase power under balanced condition. [6]
4. **Transformers:** Magnetic materials, BH characteristics, ideal and practical transformer, equivalent circuit, Transformer testing: open circuit and short circuit test, losses in transformers, regulation and efficiency. Basics of auto-transformer and three-phase transformer connections. [10]
5. **Electrical Machines:** DC Machines – construction, functions of each part, working principle, EMF and torque equations, types of dc machines. Single Phase AC machines, AC machines – classification (Split phase machines, universal machines), construction, operating principles of ac machines. [8]

Outcome: At the end of this course, students will demonstrate the ability to

- To understand and analyse basic electric and magnetic circuits.
- To study the working principles of electrical machines.

Text Books:

1. Kothari, D. P., and I. J. Nagrath, "Theory and Problems of Basic Electrical Engineering," PHI Learning Pvt. Ltd., 1998.
2. Fitzgerald, Arthur Eugene, David E. Higginbotham, and Arvin Grabel, "Basic electrical engineering," McGraw-Hill Series in Electrical Engineering, Auckland: McGraw-Hill, 1981, 5th ed.(1981).

Reference Books:

3. H.Cotton, "Advanced Electrical Technology," CBS Publication. 1967.
4. Vincent Del Toro, "Electrical Engineering Fundamentals," 2nd Edition PHI Publication. 1972.
5. Smarajit Ghosh, "Fundamentals of Electrical and Electronics Engineering", PHI Publication, 1st Edition, 2007.

EE 23.111 **Electrical Laboratory** **Total Hours: 26**

Credit: 0-0-2-1

Objective: The objective is to provide a basic exposure of installation and operation of electrical equipment.

Prerequisite: Nil

Perform any Ten of the list.

1. To Study of the various apparatus used in Electrical Engineering.
2. To perform residential house wiring using fuse, indicator, switch, Socket, plug and lamp.
3. To design a Tunnel wiring for illuminating three lamps.
4. To conduct the open circuit tests on the given single phase transformer and to determine equivalent circuit parameters.
5. To conduct the short circuit tests on the given single phase transformer and to determine equivalent circuit parameters.
6. To construct a fluorescent tube and to measure voltage across tube rod, voltage across choke and current.
7. To conduct the polarity test on the given transformer.
8. To control the speed of DC machine by field current control method.
9. To control the speed of DC machine by Armature Voltage control method.
10. To measure three phase power by using two wattmeter methods.
11. To verify Superposition Theorem using resistive circuit.
12. To verify Thevenin's Theorem using resistive circuit.

Outcome: Students will be convergent with installation and operation of electrical equipment.

MA 23.101 **Mathematics I** **Total Lectures: 40**

Credit: 3-1-0-4

Objective: To acquire fundamental knowledge of Mathematics and apply in engineering disciplines.

Pre-requisite: (10+2) Higher Secondary Mathematics

1. **Differential Calculus I:** Review of basic concepts, partial differentiation, different chain rules, second and higher order partial derivatives, Euler's theorem (with proof) on homogeneous functions and applications, Jacobians, properties of Jacobians. [07]

2. **Differential Calculus II:** Maxima and minima of functions of two variables, asymptotes (parallel and oblique), Tracing of simple Cartesian curves and standard polar curves. [07]

3. **Integral Calculus:** Basics of Beta and Gamma functions, Length of plane curves [10]

   (Cartesian and polar), area bounded by the curves (Cartesian and polar), multiple

   integrals (double and triple), change of order of integration, physical interpretations and applications.

4. **Matrix Theory**: Rank of matrix(elementary row transformations), linear dependence and independence of vectors, Solution of system of simultaneous linear equations, Eigen values, Eigen vectors of a matrix, Cayley-Hamilton theorem (without proof) and applications, diagonal and quadratic form of matrix.. [08]

5. **Three-dimensional Geometry:** Direction cosines and direction ratios of a straight line, equation of a plane, plane passing through intersections of two planes, equations of straight line, shortest distance between two skew lines, equation of a sphere, intersection of plane and sphere. [08]

Outcome: After completion of the course, students would be able to apply mathematical knowledge in engineering and solve curriculum problems.

Text Books:

1. Ramana B.V., Higher Engineering Mathematics, McGraw Hill Education, July, 2017.
2. Bali N. P. and Goyal Manish , A Text Book of Engineering Mathematics, Ninth Edition, Laxmi Publication, January 2016..

Reference Books:

1. Erwin Kreyszig Advanced Engineering Mathematics, 9th Edition, Reprint 2011.
2. Pal and Bhunia, Engineering Mathematics, Oxford University Press, 2015.
3. Dass and Verma, Higher Engineering Mathematics, S.Chand, 2019.

MA 23.102 **Mathematics II** **Total Lectures: 40**

Credit: 3-1-0-4

Objective: To acquire fundamental knowledge of Mathematics and apply in engineering disciplines.

Pre-requisite: Mathematics I (MA 23.101)

1. **Computational Methods:** Calculus of finite differences, Forward, Backward and Shift operators and their relationship, Fundamental theorem of difference calculus (without proof), Factorial function, Interpolation schemes for equal intervals (Newton's forward and backward interpolation formulae), divided difference, interpolation for unequal intervals (Newton's divided difference and Lagrange interpolation formulae), Approximation of one real root of Algebraic and Transcendental equations by Newton-Raphson method.. [12]
2. **Ordinary Differential Equations I:** Definition, order and degree of differential equation, first order and first degree differential equations (variable separable, homogeneous, reducible of homogeneous, linear, Bernoulli and exact differential equations). [07]
3. **Ordinary Differential Equations II:** Linear differential equations of second order with constant and variable coefficients, general and particular solutions (complementary functions and particular integrals), and applications. [07]
4. **Partial Differential Equations:** Formation of partial differential equations, Lagrange's multipliers method for solution of first order differential equations, separation of variables for nonlinear differential equations(general and particular solutions) and applications. [06]
5. **Vector Calculus:** Vector and scalar point functions, conservative fields, gradient of scalar point functions, divergence and curl of vector point functions (Solenoidal and irrotational vectors), directional derivative of scalar point functions, line integral (work done), surface integral, volume integral, interpretations and applications. [08]

Outcome: After completion of the course, students would be able to apply mathematical knowledge in engineering and solve curriculum problems.

Text Books:

1. Ramana B.V., Higher Engineering Mathematics, McGraw Hill Education, July, 2017.
2. Bali N. P. and Goyal Manish , A Text Book of Engineering Mathematics, Ninth Edition, Laxmi Publication, January 2016.

Reference Books:

1. Erwin Kreyszig Advanced Engineering Mathematics, 9th Edition, Reprint 2011.
2. Pal and Bhunia, Engineering Mathematics, Oxford University Press, 2015.
3. Ram Babu, Engineering Mathematics, Pearson Education, First Edition, January 2009.

ME 23.101 **Applied Mechanics** **Total Lectures: 40**

Credit: 3-1-0-4

Prerequisite Basic Mathematics and Physics

Objectives:

- This course will help the student to acquire knowledge of basic engineering mechanics concepts and enhance their ability to solve open ended engineering problems.
- To provide the student with a clear and thorough understanding of the theory and applications of engineering mechanics, covering both statics and dynamics.

1. **Introduction:** Idealization of Mechanics, concept of Rigid Body, External Forces, Laws of Mechanics. [12]
   Force Systems and Equilibrium: Force systems, Moment of force, Moment of a couple, Equivalent system, Free Body Diagram, Equations of equilibrium and their applications to various systems of forces.
2. **Distributed Forces and Moment Of Inertia:** Centroid of Composite figures, Area Moment of Inertia, Mass Moment of Inertia, Principle axes and Principle Moment of Inertia. [7]
3. **Friction:** Introduction of friction, Laws of friction, wedge, screw, belt, rolling friction. [7]
4. **Tension, Compression and Shear:** Normal stress and strain, Linear elasticity, Hooke's Law and Poisson's Ratio, Shear stress and shear strain, [8]
   Beams: Different support & load conditions, SFD, BMD
5. **Kinematics of Particles:** Introduction, Rectilinear motion, Plane curvilinear motion**,** Rectangular coordinates, Tangential and normal coordinates, Relative motion **Kinetics of Particles:** Introduction, Force, Mass and acceleration, Newton's second law of motion, Equation of motion and solution of problems, Work and Energy, Impulse and Momentum. [6]

Outcome: The students shall develop the ability to identify, formulate, and solve engineering problems related to forces and motion.

Text Books:

1. Hibbeler, Gupta R. C., "Engineering Mechanics," 11th ed., Pearson India, 2010.
2. Merian, J.L, Kraige, L.G., "Engineering Mechanics Vol I & II," 5th edition, Wiley India 2010.

Reference Books:

1. Timoshenko, S, Young, D.H, Rao, J.V, "Engineering Mechanics," 4th edition TMH, 2010.
2. Irving. H. Shames, "Engineering Mechanics," Prentice Hall Book Company, 1966.

ME 23.111 **Computer Aided Engineering Drawing** **Total Hours: 26**
Credit: 0-0-4-2

Objective:

- To improve the visualization skills.
- To enable the students with various concepts like dimensioning, conventions and standards related to working drawings, projections in order to become professionally efficient.
- To impart the knowledge on understanding and drawing of simple engineering products.

1. **Introduction to Computer Aided Sketching:** Introduction, Drawing Instruments and their uses, BIS conventions, Lettering, Dimensioning and free hand practice. Board Drawing vs. Computer- Aided Drawing. Introduction to Solid works Environments. Computer screen, layout of the software, standard tool bar/menus and description of most commonly used tool bars, navigational tools. Co-ordinate system and reference planes. Definitions of HP, VP, RPP &LPP. Creation of 2D/3D environment. Selection of drawing size and scale. Commands and creation of Lines, Co-ordinate points, axes, dimensioning, line conventions, material conventions and lettering. [8]
2. **Orthographic Projections** : Horizontal planes, Vertical planes, Front view, Top view, Side view, Projections, First and Third angle of projection, Projections of points, Principles of orthographic projections, Sketching of different views of given objects. Introduction to sketching in Solid works : 2-Dimenional Sketching, Sketching in Solid works, poly-lines, square, rectangle, polygons, splines, circles, ellipse, text, move, copy, off-set, mirror, rotate, trim, extend, break, chamfer, fillet, curves, constraints viz. tangency, parallelism, inclination and perpendicularity. [12]
3. **Projections of Lines, Planes and Solids:** Lines inclined to both the principal planes, projection of Lamina inclined to both principal planes, Projections of solids: Axis inclined to both the reference planes and Intersection of solids. Introduction to part module in Solid works: Extrude, Revolve, Sweep, Hole, Filleting, and Chamfering features. [12]
4. **Sections of solids and sectional orthographic views:** Sectional orthographic projections - Full, Half, Partial, revolved and removed surfaces. Introduction to drawing module in Solid works: Standard 3 views, Auxiliary view, Section View, Projected view. [12]
5. **Isometric Projection:** Terminology, Isometric scale, Box method, Coordinate or offset method, Four-center method, isometric projection of arcs. Solid models practice using Solidworks part module. [8]

Outcomes:

1. Students will be able to improve their visualization skills so that they can apply these skills in developing new products.
2. Students will be able to know and understand the conventions and the methods of engineering drawing using Solid works.
3. Students will be able to prepare simple drawings of engineering products.

Text Books:

1. N.D. Bhatt, "Engineering Drawing", Charotar Publishing House, 2013.
2. Prof. Sham Tickoo, Sandeep Prem Dass “Solid works 2013 for Engineers and Designers” Wiley Publication, 2013.

Reference Books:

1. Randy H. Shih, “Solid works 2013 and Engineering Graphics: An Integrated Approach”, SDC Publications. 2013

ME 23.112 **Mechanical Workshop** **Total Hours:26**

Credit: 0-0-2-1

Objectives:

1. Practice on manufacturing of components using workshop trades including fitting, carpentry, sheet metal and welding.
2. Identify and apply suitable tools for machining processes including turning, facing, thread cutting and tapping.
3. Study and practice on machine tools and their operations.

List of Jobs:

1. **Fitting Shop:** Prepare job that contain various fitting operation like filing, marking, measuring, cutting, drilling etc.
2. **Carpentry Shop:** Prepare job that contain various carpentry operations and making of carpentry joint (T joint).
3. **Sheet Metal Shop:** Prepare job that contain various sheet metal operations and making job from sheet (funnel, tray etc).
4. **Machine Shop:** Perform various operations like, turning, step turning, facing, chamfering, knurling etc. on Lathe.
5. **Welding Shop:** Prepare a weld bead using arc welding process.

Outcomes:

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

1. To acquire skills in basic engineering practice.
2. To identify the hand tools and instruments.
3. To acquire measuring skills.
4. To provides the knowledge of job materials in various shops.
5. Students will be able to analyze the material on the basis of their properties and thus assigning different weight age to their use for technical purposes.

Text Books:

1. Hajara Chaudhary, Workshop Technology Vol I & II, Media Promoters and Publishers PVT. LTD, Mumbai.
2. B.S. Raghuvanshi, Workshop Technology, Dhanpat Rai & Sons. New Delhi.

Reference Books:

1. H S Bawa, Workshop Technology, TMH. New Delhi.

PY 23.101 **Engineering Physics** **Total Lectures: 40**

Credit: 3-0-0-3

Pre-requisite: Fundamental knowledge of Basic Physics

Objective:

1. To provide students with a basic understanding of the Physics that may be required by engineers in the course of their careers
2. To enhance knowledge related to lasers and its different components to make it suitable for various purposes
3. To introduce the learners to the basics of modern physics and Quantum Mechanics

1. **Modern Physics**: Introduction to Black body radiation spectrum, Photo-electric effect, Compton effect. Wave particle Dualism. de Broglie hypothesis – de Broglie wavelength. Davisson and Germer Experiment. Matter waves and their Characteristic properties, Phase velocity, group velocity and Particle velocity, Relation between phase velocity and group velocity, Relation between group velocity and particle velocity. [8]
2. **Quantum Mechanics:** Heisenberg's uncertainty principle and its physical significance. Application of uncertainty principle (Non-existence of electron in the nucleus, explanation for decay and kinetic energy of electron in an atom). Wave function, Properties and physical significance of wave function, Probability density and Normalization of wave function. Setting up of one-dimensional time independent Schrodinger wave equation. Eigen values and Eigen functions. Application of Schrodinger wave equation – Energy Eigen values for a free particle. Energy Eigen values and Eigen functions for a particle in a potential well of infinite depth. [7]
3. **Electrical Conductivity in Metals**: Free–electron concept. Classical free-electron theory – Assumptions. Drift velocity, Mean collision time and mean free path. Relaxation time. Expression for drift velocity. Expression for electrical conductivity in metals. Effect of impurity and temperature on electrical resistivity of metals.
   Quantum free-electron theory – Assumptions. Fermi – Dirac Statistics. Fermi-energy, Fermi factor. Density of states (No Derivation). Expression for electrical resistivity / conductivity. Temperature dependence of resistivity of metals. electron in a periodic potential, origin of the energy gap, band theory of solids, distinction of solids into metals, semi-conductors and insulators. [8]
4. **Dielectric & Magnetic Properties of Materials**: Dielectric constant and polarization of dielectric materials. Types of polarization. Equation for internal field in liquids and solids (one dimensional). Ferro and Piezo electricity. Frequency dependence of dielectric constant. Important applications of dielectric materials. Classification of dia, para and ferro-magnetic materials. Hysterisis in ferromagnetic materials. Soft and Hard magnetic materials. Applications. [7]
5. **Lasers and Optical Fibers**: Einstein's coefficients (expression for energy density). Requisites of a Laser system. Condition for laser action. Principle, Construction and working of CO2 laser and semiconductor Laser. Applications of Laser – Laser welding, cutting and drilling. Propagation mechanism in optical fibers. Angle of acceptance. Numerical aperture. Types of optical fibers and modes of propagation. Attenuation. Applications–Block diagram discussion of point to point communication. [10]

Outcome: After completion of this course, students will be able to :

1. Understand the experimental evidence supporting the de Broglie hypothesis and the wave-particle duality of matter.
2. Explain fundamentals of quantum mechanics and apply to one dimensional motion of

particles.
3. Describe the band theory of solids.
4. Understand material properties to use in engineering applications and studies. 5. Illustrate the working of lasers and optical fibers.

Text Books:

1. S. O. Pillai, "Solid State Physics", New Age International, Sixth Edition, 2009
2. V. Rajendran, "Engineering Physics," TMH, Fifth Edition , 2013

Reference Book:

1. G. K. Shivkumar, "Engineering Physics," Prism Book Pvt. Ltd.
2. Stephen T. Thornton, "Modern Physics for Scientists and Engineers",Cengage Learning, 4th ed.,2013

PY 23.111 **Physics Laboratory** **Total Hours: 26**

Credit: 0-0-2-1

Pre-requisite: Nil

Objective:
1. To gain practical knowledge by applying the experimental methods to correlate with the Physics theory.
2. To apply the analytical techniques and graphical analysis to the experimental data.

List of Experiments:

1. To study of material constant, temperature coefficient of current and planck's constant.
2. To determine the specific resistance of a given wire by using the Carey Foster's bridge.
3. To measure the numerical aperture of an optical fiber.
4. To measure the cylindrical object of diameter with help of Screw gauge and Vernier caliper.
5. To study the variations of resistivity with temperature and hence determines the energy band gap of the given semiconductor with help of four probe method.
6. To convert a galvanometer into a voltmeter of a given range.
7. To convert a galvanometer into an ammeter of a given ranges.
8. To plot graph showing the variation of magnetic field with distance along the axis of a circular coil carrying current and to estimate from it the radius of the coil.

Outcome:
1. Students will be convergent with operation of equipment and verify the principle of Physics by doing the experiments.
2. Understand the concepts and principles related to material constants, temperature coefficients, and Planck's constant.
3. Analyze the collected data to determine the specific Resistance using Carey Foster's Bridge.
4. Perform measurements and data analysis to determine the numerical aperture of the optical fiber accurately.
5. Gain hands-on experience in measuring and plotting the magnetic field variation along the axis of a circular coil carrying current.
6. Develop the ability to convert a galvanometer into an ammeter and voltmeter.

UC 23.102 **Pragmatic English Communication** **Total Lectures: 26**

Credit: 2-0-0-2

Objective: This course aims to develop and enhance the linguistic and communicative competence of the students. The students will be exposed to various forms of personal and professional communication. The self-learning tasks designed will facilitate to enhance effective communication skills.

Pre-requisite: Basic knowledge of English grammar.

1. **Sentence Structure:** Elements of a sentence (Subject, verb, object, complement and adjunct) [7]
   Correct Usage of Language: Tenses (Present, Past and Future, Modals (can, could, will, would, shall, should, may, might, must, ought to, need).
2. **Listening and Reading Skills:** Developing effective listening and reading. [6]
   Writing Skills: Theme (Paragraph) writing, Letter (formal letters) and application writing, report writing, résumé writing
3. **Idioms and Phrasal Verbs:** Understanding idioms, collocations, introduction to phrasal [5]
4. **Common Errors:** How to avoid common sentence – structural errors, vocabulary and pronunciation [3]
5. **Technology-based Communication:** Effective e-mail messages; power-point presentation; enhancing editing skills using computer software. [5]

Outcome: The students will learn the essentials of English Grammar, analyze, identify and rectify the common errors in speech and writing. Student can read and write with ease and fluently, enrich vocabulary and enjoy reading and writing.

Text Books:

1. Wren & Martin: English Grammar & Composition, S. Chand & Co, Delhi (Latest edition)
2. Raman, M., & Singh, P. (2017). Business Communication (2nd edition.). New Delhi: Oxford University Press.

Reference Books:

1. Prasad, H. M. How to Prepare for Group Discussion and Interview. New Delhi: Tata McGraw-Hill Publishing Company Limited
2. Bonet, Diana. The Business of Listening: Third Edition. New Delhi: Viva Books, 2004.
3. Hornby A.S.: A Guide to patterns and usage, Oxford University Press, Delhi, 1954.
4. Lesikar, Raymond V and Marie E. Flatley. Basic Business Communication: Skills for Empowering the Internet Generation: Ninth Edition. New Delhi: Tata McGraw-Hill

CS 23.201 **Discrete Mathematics** **Total Lectures: 40**

Credits: 3-1-0-4

Prerequisite(s): None

Objective(s): To Introduce fundamental concepts and applications of discrete structure in the field of computer science

1. **Sets, Relations and Functions**: Set theory, combination of sets, multi-sets, ordered pairs, set identities, relations, operations on relations, properties of relations, composite relations, equality of relations, order of relations, functions, classification of functions, operations on functions, recursively defined functions. [8]
2. **Partially Ordered Sets**: Definition, combination of partial order sets, Hasse diagram, lattices: definition, properties of lattices: bounded, complemented, modular and complete lattice, morphisms of lattices. [6]
3. **Algebraic Structures & Counting:** Groups, subgroups and order, cyclic groups, cosets, Lagrange's theorem ,normal subgroups, permutation and symmetric groups, group homomorphisms, rings, fields, basics of linear algebra ,integers modulo n. Counting principles, permutations, combinations, summations, principle of inclusion and exclusion, pigeon-hole principle, generating functions and recurrence relations. [10]
4. **Propositional Logic:** Proposition, well formed formula, truth tables, tautology, satisfiability, contradiction, algebra of proposition, theory of inference, natural eduction, predicate logic: first order predicate, well formed formula of predicate, quantifiers, and inference theory of predicate logic. [8]
5. **Graph Theory:** Graphs, types of graphs, subgraphs, Eulerian chains and cycles, Hamiltonian chains and cycles, graph traversal algorithms, hromatic number, plannar graph. Tree, Spanning tree, minimum spanning tree algorithms. [8]

Outcome(s): On the successful completion of the course, the student will be able to:

- Understand the theory and techniques of logic, graphs and trees, and algebraic systems.
- Apply the knowledge and skills obtained to investigate and solve a variety of discrete mathematical problems.

Text Books:

1. Bernard Kolman, Robert C., BusbSharon, Cutler Ross, Discrete Mathematical Structures,PHI, 6th Edition,2009.
2. Kenneth H. Rosen, Discrete Mathematics and Its Applications, TataMcGraw-Hill,7th Edition,2011.

Reference Books:

1. Swapan Kumar Chakraborty, Bikash Kanti Sarkar, Discrete Mathematics, Oxford Higher Education,2011
2. C.L. Liu, D.P. Mohapatra, Elements of Discrete Mathematics A Computer Oriented Approach,TataMcgraw-Hill,3rd Edition,2008.
3. Seymour Lipshutz, Marc Lipson, Discrete Mathematics, 3rd edition, 2010

CS 23.202 **Advanced Programming** **Total Lectures: 40**

Credit: 3-0-0-3

Prerequisite(s): CS 23.101(Programming for Problem Solving)

Objective(s): The objective of this course is to:

- Cover issues related to the definition, creation and usage of classes, objects and methods.
- Discuss the principles of inheritance and polymorphism.
- Provide the foundation of good programming skills by discussing key issues of the design of object- oriented software, including programming design and documentation.
- Cover the basics of API and to explore the java abstract programming interface.

1. **Overview:** Introduction to OOP, OOP concepts, interactive Development Environments: Eclipse/Net Beans, basics of OOP, importance of Java Primitive data types, type casting, arrays, operators and control structures, identifiers, constants and literals. [8]
2. **Classes and Inheritance**: Class fundamentals, objects, constructors, this keyword, garbage collection, the finalize()method, overloading methods, understanding static &final, exploring the string class. Inheritance basics, using super, multi level hierarchy, constructor in hierarchy, method overriding, dynamic method dispatch, the object class, using abstract classes, interface, I/O with different types of streams(Character, byte and primitive data type), string handling with String and String Buffer. [8]
3. **Package, and Exception Handling:** Packages, access protection, importing packages,interfaces,exception-handlingfundamentals,exceptiontypes,usingtry and catch, multiple catch clauses, nested try statements, throw, throws, finally, Java's built-in exceptions, user defined exception, assertions [8]
4. **GUI and Multi threaded Programming and JDBC**: Swing Introduction and GUI Widget, form design with swing, Graphics with swing, The Java thread model, the main thread, creating multiple threads, threads priorities, managing threads, thread class built-in functions. Connection to MySQL and SQL operations. [8]
5. **Java Collections:** Introduction, Types of Collections, Dynamic array, Linked list, Stack, Hash table, TreeSet, Abstract List, Queue, Abstract Sequential List, Linked Hash Set, Priority queue, Double-ended queue with applications and operations. [8]

Outcome(s): Upon completion ofthis course, students areableto:

- Understand the concepts of OOP as well as the purpose and usage principles of inheritance, polymorphism, encapsulation and method overloading.
- Identifyclasses, objects,members ofa classand the relationships amongthem needed for aspecific problem.
- Develop programs using the in-built JavaAPIandthe Javastandard class library.Become able to Implement Data Structure, Graphics and CURD operations in Java
- CreateJavaapplicationprogramsusingsoundOOPpractices(e.g.,interfacesandAPIs) andproperprogramstructuring(e.g.,by usingaccesscontrolidentifies,automatic documentation throughcomments,

error exception handling).

- Useof development environment like Eclipse/NetBeans/InttelliJforcollaborativeprogramming

Text Books:

1. Herbert Schildt, Java The Complete Reference, Herbert Schildt, 8th Edition, 2011.
2. Bert Bates and Kathy Sierra, Head First Java, O'Reilly,2005

Reference Books:

1. Allen B. Downey, Think Java:Howto ThinkLikeaComputer Scientist, 2011.
2. Paul Deitel, HarveyDeitel,JavaHowTo Program (EarlyObjects),10th Edition, 2014.
3. E. Balaguruswamy, Programing with Java A Primer, Tata Mc. Graw Hill, 2007

CS-23.203 **Data Structures** **Total Lectures: 40**

Credit:3-0-0-3

Prerequisite(s): CS 23.101 (Programming for Problem Solving-I)

Objective(s): Toprovideknowledgeofwaysofstructuringandoperatingondata,thenatureofsome fundamentalproblems,methodsfor addressingthoseproblems,andtopromote an analytical and empiricalappreciation ofthe behavior of algorithmsusingdata structures

1. Introduction: Basic terminology, elementary data organization, algorithm, efficiency of an algorithm, time and space complexity, asymptotic notations, time-space trade-off, abstract data types. [4]
2. Array&Linkedlist: Definition, single and multidimensional arrays, representation of arrays: row major order, and column majororder,application of arrays, sparse matrices and the irrepresentations. Singly linked lists, doubly linked list, circular linked list, insertion, deletion, traversal, polynomial representation and addition, generalized linked list. [10]
3. Searching &Sorting: Sequential search, binary search, comparison and analysis, hash function, collision resolution strategies, insertion sort, selection sort, bubble sort, quick sort, mergesort,heap sort, radix sort. [10]
4. Stack& Queue: Primitive stack operations : push and pop array and linked list implementation of stack, application of stack: prefix and postfix expressions, evaluation of postfix expression, recursion, and tower of Hanoi problem. Queue operations, circular queues, array and linked list implementation of queues, dequeue and priority queue [8]
5. Tree: Binary trees, binary tree representation, tree traversal algorithms: inorder, preorder and postorder, threaded binary trees, Huffman algorithm, binary search trees AVL trees. [8]

Outcome(s): On the successful completion of the course, the student will be able to:
beknowledgeable ofcertain abstract data types (ADT)
be able to makea critical assessment of different implementations of an ADT
be familiarwith anumber offundamentalcomputational problems, and beaware of real world instances ofthose problems

Text Books:
1.Lipschutz, “DataStructures”Schaum’sOutlineSeries, TMH, Revised Edition Feb. 2014
2. Horowitzand Sahani, Fundamentals of DataStructures, University Press,2ndEdition, 2008.

Reference Books:
1.AaronM.Tenenbaum,YedidyahLangsamandMosheJ.Augenstein,DataStructures UsingC and C++, PHI, 2ndEdition, 2012.
2. D. Samanta, Classic Data Structures, PHI, 2nd Edition, 2009
3. JeanPaulTrembleyandPaulG.Sorenson,AnIntroductiontoDataStructures withapplications, McGraw Hill, 2ndEdition, 2003.

CS-23.204 **Database Management Systems** **Total Lectures: 40**

Credit:3-0-0-3

Prerequisites: CS 23.203 (Data Structures).

Objective(s): Theobjectiveof this Courseis to:

- Understand thefundamental concepts of databasemanagement.
- Designand Normalizedatabases.
- Understand transaction processingcontrol.
- Understand basicof object concepts and object relations.
- Understand theroleof theDBMS&RDBMSin theorganization.

1. **Introduction to database systems:** database system concepts and architecture,datamodeling using the**Entity-relationship(ER)model**.MappingER modelto relational model.**RelationalDataModel:**relationalmodelconstraints,relationaldatabase schemas,updateoperations,dealing withconstraintviolations,relationalalgebra operations, and relationalcalculus. [10]
2. **StructuredQueryLanguage(SQL)**:SQLdatadefinitionanddatatypes,specifying constraints,schemachange statements, DDL,DML, basicqueries,complexqueries,update statements, views, Joins, Inbuilt-Functions of SQL [6]
3. **DatabaseDesign:**Functionaldependenciesandnormalizationofrelationaldatabases , firstnormalform(1NF), 2NF, 3NF, boyce-coddnormalform(BCNF), multivalued dependency, 4NF, join dependency, 5NF. [8]
4. **Transaction Processing:** Introduction-Transaction State, Transaction properties, Concurrent Executions, need of serializability, conflict v/s view serializability, testing for serializability, recoverable schedules, cascadeless schedules.Concurrency controltechniques:twophaselocking protocol;concurrencycontrolbased ontime-stampordering, Databaserecoverytechniques:recoveryconcepts,recovery techniques based on deferred and immediateupdate [10]
5. **Database Security**:Database Security**:**Vulnerabilities in database, authentication, SQL injections, Encryptions & public Key infrastructures, privacy issues and preservation process [6]

Outcome(s): After completion ofthecourse, students willbe able to:

Analyze database models & entity relationship models. Understand architecture for database development.

Design E-R diagram

Use Structured Query Language (SQL) with complex queries.

Design and normalize the relations for any organization.

Understand transaction processing control.

Text Books:

1.RamezElmasri&ShamkantB.Navathe,FundamentalsofDatabaseManagementSystems, Edition, Pearson Education, 5th ed ,2013.

2. Raghu Ramakrishnan, DatabaseManagement Systems, TataMcGraw Hill3rdEdition,2008.

Reference Books:

1. G.W. Hansen, and J.V. Hansen, Database Management and Design, 2nd Edition, Prentice Hall India, 2000.
2. Korth and Schilberschatz, Database System Concepts, 5th Edition, Tata McGraw Hill, 2006.
3. C.J. Date, Database Systems, Tata McGraw Hill, 2nd Edition, 2003.

CS 23.205 **General Purpose Programming** **Total Lectures: 40**

Credit: 3-0-0-3

Prerequisite(s): CS 23.101 (Programming for Problem Solving -I)

Objective(s): Python is a general-purpose, versatile and popular programming language. It's great as a first language because it is concise and easy to read, and it is also a good language to have in any programmer's stack as it can be used for everything from software development and scientific applications.

1. **Introduction**: The basic elements of python, python syntax, python operators, basic input, branching programs, control structures, data types, strings, tuples, lists, sets, dictionaries, bytes, bytearray and input iteration, Mutability with different types, ranges and iterators [7]
2. **Functions and Module**: Functions and scoping flow of execution, specifications, recursion, global variables, variable arguments, default arguments, system functions and parameters, input and output, modules , making your own module, importing modules, decorators, generators, command line argument. [7]
3. **Classes and Object-Oriented Programming**: Abstract data types and classes, class attribute, instances, object overhead, constructor, self, inheritance, encapsulation and information hiding, polymorphism, getter & setter, abstract class, duck typing, monkey patching, operator overloading, method overloading using multiple dispatch decorator. [8]
4. **Testing and Debugging**: Exceptions and assertions, types of testing – black-box and glass-box, debugging, handling exceptions, assertions. [6]
5. **Advance Python:** Regular expressions – REs and Python, plotting using matplotlib and Multithreaded Programming, GUI & event handling programming using Tkinter & pygame. Introduction to machine learning and deep learning [12]

Outcome(s): Upon completion of this course students will be

- Familiar with Python syntax.
- Able to use python to implement a given logic.

Text Books:

- John V Guttag, Introduction to Computation and Programming using Python, Prentice Hall of India, 2nd Edition, 2013.
- Martin C. Brown, Python: The Complete Reference,Mc Graw Hill, 2018

Reference Books:

- Zed A. Shaw ,Learn Python the hard way: a very simple introduction to the terrifyingly beautiful world of computers and code, 3rd edition, 2014.
- Richard L. Halterman,Learning to Program with python, 2011.
- Allen Downey ,Think Python How to Think Like a Computer Scientist, Version 2.0.17, Green Tea Press Needham, Massachusetts,2012.

CS 23.206 **Theory of Computation** **Total Lectures: 40**

Credit:3-0-0-3

Prerequisite(s): CS23.201(Discrete Mathematics)

Objective(s): The goal of this course is to

- Provide students with an understanding of basic concepts in the theory of computation.
- To understand variety of issues in the mathematical development of computer science theory, particularly finite representations for languages and machines, as well as gain a more formal understanding of algorithms and procedures.

1. **Finite Automata and Regular Expression:** Finite state systems, basic definitions, regular languages and regular expression, finite automata ,Equivalence of DFA and NDFA , Kleen's theorem, pumping lemma for regular sets, regular expression with algebraic method Using Arden's Theorem, closure properties of regular sets, minimization of finite automata, Mealy and Moore machine - conversion of Mealy machine into Moore machine and vice versa. [11]
2. **Context-Free Grammars:** Context-free grammars, union, concatenation of CFG, derivation trees, simplification of context-free grammars, Chomsky normal form, Greibach Normal form, ambiguous context-free grammar. [9]
3. **Context-Free Languages and Pushdown Automata:** Pushdown automata and context-free languages, pumping lemma for CFL, closure properties of CFL, decision, algorithms for CFL.. Equivalence of PDA and CFL. [8]
4. **Turing Machine:** Turing machine model, techniques for Turing machine construction, Turing machines as language acceptor, components and description of TM, properties of recursive and recursively enumerable languages, universal Turing machine. [8]
5. **Chomsky Hierarchy of Languages:**Four types of Languages and grammars, Regular grammars, unrestricted grammars and Turing machines, context-sensitive grammars and linear bounded automata. [4]

Outcome(s):

- Be ableto provetheequivalenceof languages described byfinite state machines and regularexpressions.
- Be ableto construct pushdown automata and theequivalent context freegrammars.
- Be ableto provetheequivalenceof languages described bypushdown automataand context freegrammars.
- Be ableto construct Turingmachines and Post machines.
- Be ableto provetheequivalenceof languages described byTuringmachines and Post Machines.

Text Books:

1.Peter Linz, An Introduction to Formal Languages and Automata, Third Edition, Jones and Bartlett, 2016.

2. Hopcroft, Motwani, Ullmann, Introduction to Automata Theory, Languages and Computation, 2nd Edition, Pearson Education, 2005.

Reference Books:

1. Daniel I. A. Cohen, Introduction to Computer Theory, Prentice-Hall, 2nd Edition,1997.

2. Martin,Introduction to Languagesand the theoryofComputation, 3rdEdition,Tata McgrawHill2006.

3. Michael Sipser, Introduction to the Theory of Computation, 3rdEdition, Cengage Learning,2013

CS 23.208 **Computer Organization and Architecture** **Total Lectures: 40**

Credit:3-0-0-3

Prerequisite(s): EC 23.202 (Digital Electronics) (Optional)

Objective(s):

- To have a thorough understanding of the basic structure and operation of a digital computer.
- To discuss in detail the operation of the arithmetic unit including the algorithms &
- implementation of fixed-point and floating-point addition, subtraction, multiplication & division.
- To study the different ways of communicating with I/O devices and standard I/O interfaces. To study the hierarchical memory system including cache memories and virtual memory.

1. **Introduction:** Von-Neuman architecture, register transfer and micro operations: register transfer language, arithmetic micro-operations, logic micro-operations, shift micro- operations, bus and memory transfers, computer organization and design: instruction cycle, computer registers, common bus system, computer instructions, addressing modes, design of a basic computer. General register organization, stack organization, instruction formats, data transfer and manipulation, program control, RISC, CISC [10]
2. **Computer Arithmetic:** Hardware implementation of addition and subtraction, multiplication algorithms, Booths multiplier, array multiplier, division algorithms, floating point arithmetic: floating point representation, add, subtract, multiplication, division. [7]
3. **Memory & I/O Organization:** Memory hierarchy, RAM, ROM chips, memory address map, memory connection, associative memory, cache memory and virtual memory: paging and segmentation, auxiliary memory: magnetic disks, magnetic tapes, Input-output interface, Asynchronous data transfer, modes of transfer, priority interrupt, DMA, IOP. [8]
4. **Hardwired and Microprogrammed Control:** Hardwired control, microprogrammed control: microinstruction, microprogram sequencing, wide-branch addressing, microinstruction with next-address field, prefetching microinstruction. [7]
5. **Pipelining & Parallel Processing:** **P**ipeline and vector processing: pipeline structure, speedup, efficiency, throughput and bottlenecks, arithmetic pipeline and instruction pipeline. Flynn's Classification performance matrices for parallel system, Contemporary issues: Recent Trends: Multiprocessor architecture: Overview of Shared Memory architecture, Distributed architecture. [8]

Outcome(s): On the successful completion of the course, the student will be able to:

- Understand basic structure of a digital computer.
- Perform arithmetic operations of binary number system.
- Know the organization of the control unit, arithmetic and logical unit and I/O memory unit

Text Books:

1. M. Morris Mano, Computer System Architecture, Pearson Education, 3rd edition, 2008.
2. Carl Hamacher, ZvonkoVranesic and SafwatZaky, Computer Organization, Tata McGraw Hill, 5th edition, 2002.

Reference Books:

1. Patterson and John L. Hennessy, Computer Organization and Design: The Hardware/ Software interface, Elsevier, 3rd edition, 2005.
2. William Stallings, Computer Organization and Architecture – Designing for Performance, Pearson Education, 6th edition, 2000.
3. Kai Hwang, Advance Computer Architecture,Tata Mc Graw Hill,18th edition,2008

CS-23.211 **Software Engineering** **Total Lectures:40**

Credit:3-0-0-3

Prerequisite(s): CS-23.101 (Programming for Problem Solving -I)

Objective(s):

- To understand the basic software engineering approaches, software development process models, project planning, software size estimation, software effort and cost estimation and risk management.
- To know about software requirements specification, requirement elicitation, structured system analysis and introduction of object-oriented analysis and design, design techniques.
- To acquire knowledge on implementation issues such as modularity and coding standards.

1. **Introduction:** Programs versus software products, software crisis, need for software engineering, attributes of effective software, software myths, software characteristics, and role of management in software development, overview of quality standards like ISO 9001, SEI-CMM. [6]
2. **Software Processes and Requirement Analysis:** Software engineering: a layered technology, the software process, software process models, component based development, problem analysis, requirement engineering, requirement elicitation, requirement analysis, software requirement and specifications, data flow diagrams, data dictionaries, entity-relationship diagrams. [8]
3. **Software Design:** The design process, design principles, design concepts, effective modular design: cohesion & coupling, classification of cohesiveness & coupling, design notations, design tools: DFD, ER Diagram, Use Case Diagram, Class Diagram. structure charts, introduction to object-oriented design. [9]
4. **Software Project Planning:** Responsibilities & skills of a project manager, project planning, and project size estimation: LOC, function point & feature point metric, project estimation techniques, COCOMO model, staffing level estimation, scheduling, risk management, software configuration management. [8]
5. **Coding and Testing:** Software testing fundamentals, test case design, white box testing:-basis path testing, control structure testing, black box testing: boundary value analysis, equivalence class testing, decision table testing, cause effect graphing, level of testing, debugging, testing tools &standards, testing for client/server architecture, real time systems and GUIs testing. Brief of software re-use, reverse engineering, software re-engineering, agile software engineering. [9]

Outcome(s):

On successful completion of this course, students will be able to

- Demonstrate knowledge of the distinction between critical and non-critical systems and ability to manage a project including planning, scheduling and risk assessment/management.
- Author a software requirements document and understanding of the proper contents of a software requirements document.

Text Books:

- R. S. Pressman, Software Engineering – A practitioner's approach,Tata-McGraw-Hill Int. Ed., 8th edition, 2014.
- K. K. Aggarwal, Yogesh Singh, Software Engineering, New Age International, 1st edition, 2008.

Reference Books:

- Pankaj Jalote, An Integrated Approach to Software Engineering, Narosa Publisher, 3rdedition, 2005.
- Rajib Mall, Fundamentals of Software Engineering, Prentice-Hall, 2ndedition, 2004.
- Robert C. Martin, Agile Software Development, Principles, Patterns and Practices, 1stedition, 2002.

CS 23.252 **Advanced Programming Laboratory** **Total Hours: 26**

Credit: 0-0-2-1

Prerequisite: (CS23.202) Java Programming and (CS 23.102) Programming for Problem Soving -II)

Objectives: The objective of this course is to:

- Become comfortable with object-oriented programming and learn to think in terms of objects.
- Provide the foundation of good programming skills by discussing key issues to the design of object-oriented software, including programming design patterns, documentation.

Contents:

- Code different pattern to understand control structures
- Code program like factorial, fibinocci etc. to understand functions and recursions.
- To understand OOP implement an small application with getter and setter method
- Implement various operators, this keyword and constructors
- Code different usage of static keyword, polymorphism
- Designing of Forms in Swing and Connection to database
- Show different usage of String class using its inbuilt functions
- Demonstrate Java Collections and Applications
- Demonstrate exception handling using all its keywords
- Demonstrate console I/O and File I/O
- Implement multithreading using Runnable interface and Thread class

Outcomes: Upon completion of this course, students are able to:

- Understand the concepts of OOP as well as the purpose and usage principles of Inheritance, polymorphism, encapsulation and method overloading.
- Create Java application programs
- Use of development environment like Eclipse for collaborative programming/editing.

CS-23.253 **Data Structures Laboratory** **Total Hours:26**

Credit:0-0-2-1

Prerequisite(s): (CS 23.111)Programming Laboratory-I

Objective(s):

- To develop skills to design andanalyzesimple linear and nonlinear datastructures
- To Strengthen theabilityto identifyand applythesuitabledata structure forthegiven real world problem
- To Gain knowledgein practicalapplications of data structures

Contents:

- Implement various operations on arrays
- Implement different operations on different types of link list
- Implement various searching algorithms
- Implement different sorting algorithm and realize their complexities
- Implement Queue and stacks with all the operations
- Implement Binary Tree
- Implement graphs

Outcome(s): At the end ofthis course,the student will

- Be ableto design and analyzethe time and spaceefficiencyof thedata structure.
- Becapable to identitythe appropriate data structureforgiven problem.
- Havepractical knowledgeon the application ofdata structures.

CS-23.255 **General Purpose Programming Laboratory** **Total Hours:26**

Credit:0-0-2-1

.

Prerequisite: (CS 23.111) Programming Laboratory-I

Objectives: The objective of this course is to:

- Become comfortable with Python programming, starting from basics of data types to deep learning.
- Provide the foundation of to explore any expansions of Python whether it is machine learning, deep learning, web development or gaming. It will also put focus to learn various skills to design object-oriented software.

Contents:

- Code different data types and operations on list, tuple, dictionaries, sets, String etc
- Code basic programs like factorial, fibonacci etc. to understand functions and recursions.
- Implement different built-in functions in Python like eval, exec,lambda, map reduce, filter etc.
- Implement the concept of class, objects, inheritance and polymorphism.
- Implement constructor and self
- Code programs to implement exceptions and assertions
- Implement regular expressions
- Implement a small gaming application using Pygame
- Implement user interface and event handling using TKinter
- Code programs to implement the logic for machine and deep learning
- Implement multithreading

Outcomes: Upon completion ofthis course, students will be:

- Able to develop application in Python
- Able to develop scientific applications for machine learning
- Able to use python to implement a given logic.
- Able to use various Python IDE like Anaconda, Pycharm etc.

EC 23.202 **Digital Electronics** **Total Lectures: 40**

Pre-requisite: Basics of Electronics Engineering **Credits: 3-0-0-3**

Objective: The aim of this course is to present the principles and techniques of combinational and sequential digital logic design and optimization at a gate level. The use of transistors for building gates is also introduced.

1. **Number Systems and Boolean Algebra:** Introduction to number systems, weighted and non-weighted codes, 1's complement, 2's complement, complement arithmetic, Postulates and theorems of Boolean algebra, Boolean functions, canonical and standard form, simplification of Boolean function using Boolean laws and theorems. [5]
2. **Logic Gates and Simplification of Boolean Functions:** Basic Logic gates, Universal gates, K-map representation, incompletely specified functions, simplification realization with gates, Quine-Mc Cluskey method. [5]
3. **Combinational Logic:** Analysis and design of combinational circuits, half adder and full adder, half subtractor and full subtractor, binary serial and parallel adder, BCD adder, binary multipliers, magnitude comparator, decoders, encoders, multiplexers, de-multiplexers. [10]
4. **Sequential Circuits and Programmable Logic :** Latches, flip-flops, triggering of the flip-flops, master-slave flip-flop, excitation tables, conversion of the flip-flops, analysis and design of clocked sequential circuits, shift registers, counters, FPGA, logic families: I^2l, GTL, BiCMOS [10]
5. **Introduction to Verilog HDL:** Verilog as HDL, Levels of Design Description, Concurrency, Simulation and Synthesis, Functional Verification, System Tasks, Programming Language Interface (PLI), Module, Simulation and Synthesis Tools. Gate Level Modeling: Introduction, AND Gate Primitive, Module Structure, Other Gate Primitives, Illustrative Examples, Tri-State Gates, Array of Instances of Primitives, Design of Flip-flops with GatePrimitives, Delays, Strengths and Construction Resolution, Net Types, Design of Basic Circuits. [10]

Outcomes: Upon successful completion of this course, students should be able to:

- Gain fundamental knowledge of the basic logic gates such as AND, OR, NAND NOR, EX-OR and their implementation in Logic Circuits.
- Design multipliers, Combinational Circuits and registers.

Text Books:

- Mano, M. M. and Ciletti, M.D., "Digital Design", Pearson Education, Patparganj, Delhi, 4thEdition, 2009.
- Floyd, T. L. and Jain, R.P., "Digital Fundamental", Pearson Education, Patparganj, Delhi, 8thEdition, 2009.

References

- Puri, V. K., "Digital Electronics: Circuits and Systems", Tata McGraw-Hill, Patel Nagar, New Delhi, 4thEdition, 2001.
- Jain, R. P., "Modern Digital Electronics", Tata McGraw-Hill, Patel Nagar, New Delhi, 4th Ed.

EC 23.224 **Digital Electronics Laboratory** **Total Hours:26**

Credit: 0-0-2-1

Pre-requisite: Basics of Electronics Engineering

Objective: This lab course develops skills for implementing and analyzing digital electronic circuits, with applications to electronics design. It is also designed to enhance problem solving ability through steadily increasing problem complexity.

- To study the use of General Purpose Digital Trainer and familiarization with the digital IC Data-Books.
- To study AND, OR. NOT, NAND, NOR, EX-OR gates and their specified experiments.
- To set up a 4-bit Binary Counter (74LS93) to demonstrate its sequence of Counting.
- To design and test 4-bit Adder and Subtractor circuits.
- To compare the magnitude of two 4-bit Binary Numbers A and B.
- Design a combinational circuit with 4 inputs and 4-outputs that converts a 4-bit Gray Code to an equivalent 4-bit Binary number.
- Design and implement a BCD to Seven Segment Decoder / Driver.
- To design and implement a 2 to 1 multiplexer.
- To design and implement a 1:2 & 1:4 de multiplexer.
- To investigate the operation of various flip flops SR, JK, T and D using gates and FF ICs.
- To design and implement Adders and Subtractors using Verilog HDL
- To design and implement Multiplexers/De-Multiplexers using Verilog HDL
- To design and implement Flips-Flops using Verilog HDL

Outcomes:

1. The background provided in this laboratory may effectively be utilized for the understanding of advanced courses such as VLSI Design and for the designing of complex circuitry.
2. Gain a hands-on experience in carrying out key operations needed to assemble circuits on breadboard.
3. This subject is directly applicable to industrial IC design and provides a strong background for more advanced courses.
4. This lab will be useful in the designing of hardware projects using Verilog HDL

CS 23.254 **Data Base Management Systems Laboratory** **Total Hours: 26**
Credit:0-0-2-1

Pre-requisites: CS 23.204 (**Data Base Management Systems**)
Objective: By the end of the practical sessions of this section, students will be able to:

- Create Databases using a user interface and SQL command.
- Create Integrity and constraints on databases.
- Write SQL queries.
- Provide a practical overview of advanced concepts like triggers, assertion, views, etc.

Contents:

- Creation of Databases
- Implement various Integrity Constraints
- SQL queries to implement DDL
- Implement various queries to implement DML
- Implement logics for PL SQL
- Implement Triggers
- Some examples of query

Outcomes: After undergoing this laboratory module, the students should be able to:

- Understand, appreciate and effectively explain the underlying concepts of database technologies.
- Design and implement a database schema for a given problem-domain.
- Populate and query a database using SQLDML/DDL commands.
- Declareandenforceintegrityconstraintsonadatabaseusingastate-of-the-artRDBMS.
- Programming PL/SQL including stored procedures, stored functions, cursors, packages.

MA 23.201 **Mathematics III** **Total Lectures:40**

Credit: 3-1-0-4

Pre-requisite: Mathematics I and Mathematics II

Objective: To acquire fundamental knowledge of Linear programming, Numerical Methods, Probability, Statistics and Integral Transform and apply in engineeringdisciplines.

Linear Programming: Introduction to linear programming problems, Graphical solution, Simplex method, Artificial variable technique (Two phase method, Big –M method). [8]

Numerical Methods: Solution of system of equations (Gauss elimination and Gauss - Seidel methods); Solution of algebraic and transcendental equations (Bisection and Regula Falsi methods); Solution of ordinary differential equation (Euler's modified, Milne's predictor & corrector, Runge-Kutta method of orderfour).

[8]

Integral Transform: Basics of Laplace transform and inverse Laplace transform; First and Second Shifting theorems; Convolution theorem (without proof); Solution of ordinary differential equations with initial conditions by Laplace transforms.

[8]

Probability Theory: Recapitulation of basics of permutation, combination and probability theory; Law of addition and law of multiplication; Conditional probability; Total probability; Baye's theorem(withproof)andit'sapplications.

[8]

Correlation and Regression: Conceptof least square technique , fitting of least square straight line and quadratic curve (Parabola)**,** correlations analysis, different types,Simple, partial and multiple correlations, correlation of bivariate frequency distribution. [8]

Outcome: After completion of the course, students would be able to apply the knowledge of linear programming, numerical methods, statistics,probability and integral transforms in engineering and solve curriculumproblems.

TextBooks:

1. Ramana B.V., Higher Engineering Mathematics, TMH,2016.

2. P.K. Gupta and D.S. Hira, Operations Research, S. Chand, 2019.

Reference Books:

1. Sastry S.S., Introductory Methods of Numerical Analysis, 5th Edition,Pearson.

2. Jain R.K. and S.R.K. Iyenger, Advanced Engineering Mathematics, Narosa Publishing House,2003.

CS 23.301 **Design and Analysis of Algorithms** **Total Lectures: 40**
Credit:3-1-0-4

Prerequisite(s): CS-23.201(Discrete Mathematics), CS-23.203(Data Structures)

Objective(s):
- To learn the algorithm analysis techniques.
- To become familiar with the different algorithm design techniques.
- To understand the limitations of the algorithms power.

1.	**Introduction:** Fundamentals of algorithms and problem solving, asymptotic notations, master theorem, time complexity of algorithms, algorithm design paradigms and their applications in sorting and searching..	[8]
2.	**Greedymethod ,Divide and conquer andDynamic programming:**General Method**,** knapsack problem, job sequencing problem, Strassen matrix multiplication, 0/1 knapsack, assembly line scheduling, matrix chain multiplication.	[8]
3.	**Backtracking and Branch & Bound:**General Method, 8 queens problem, sum of subsets, 0/1 knapsack problem,	[8]
4.	**Tree and Graph algorithms:** Spanning tree algorithms- Prim and Kruskal's algorithm,Huffman coding, Graph Traversal- BFS and DFS, Shortest path algorithms-Dijkstra, Bellman FordWarshall's.	[8]
5.	**Complexity Classes:** P,NP,NP hard and NP complete problems, reducibility, decision and optimization problems, examples of NP complete problems - 3 CNF, clique, vertex cover and traveling salesman problem, Introduction of approximation Algorithms.	[8]

Outcome(s): On the successful completion of the course, the student will be able to:
- Design algorithms for various computing problems.
- Analyze the time and space complexity of algorithms.
- Critically analyze the different algorithm design techniques for a given problem.
- Modify existing algorithms to improve efficiency.

Text Books:
1. Thomas H.Cormen, Charles E.Leiserson, Ronald L. Rivest and Clifford Stein, Introduction to Algorithms, PHI Learning Private Limited, 3rd edition, 2016.
2. Ellis Horowitz, SartajSahni, SanguthevarRajasekaran, Computer Algorithms, Silicon Press, 2nd edition, 2008.

Reference Books:
1. Anany Levitin, Introduction to the Design and Analysis of Algorithms, Pearson Education, 3rd edition, 2012.
2. Alfred V. Aho, John E. Hopcroft and Jeffrey D. Ullman, Data Structures and Algorithms, Pearson Education, 1st edition, 2006.
3. Donald E. Knuth, The Art of Computer Programming, Volumes 1& 3, Pearson Education, 2009.

CS 23.302 **Artificial Intelligence and Machine Learning** **Total Lectures: 40**
Credit: 3-0-0-3

Prerequisite(s): (Programming for Problem Solving I), (Logic in Computer Science)

Objective(s):
- To understand the basic problem solving in Artificial Intelligence.
- To learn different blind as well as heuristic search strategies.
- To understand the various ways of knowledge representation in Artificial Intelligence.
- To know the concepts of machine learning.

1. **Overview of Artificial Intelligence:** Introduction to artificial intelligence and intelligent agents, brief history of artificial intelligence, related research fields, AI techniques, AI problems, applications and future of AI. [6]
2. **Problem Solving and Search Techniques:** Basics of problem solving, Problem representation paradigms, defining problem as a state space representation, uninformed search, depth first search, breadth first search, heuristic search, heuristic functions, generate and test, hill climbing, best first search, A* algorithm, problem reduction, AND/OR graph, AO* algorithm. [8]
3. **Knowledge Representation and Logic:** Representation and mappings, procedural versus declarative knowledge, approaches and issues in knowledge representation, first order logic, higher order logic, unification and resolution for performing logical inference, probabilistic logic. [8]
4. **Game Theory:** Minimax search, alpha-beta cutoffs, iterative deepening, references on specific games. [8]
5. **Machine Learning:** Well posed learning problems, Components of learning, supervised learning, unsupervised learning, reinforcement learning, Designing a learning system, support vector machines, KNN, learning decision trees, association rule mining, Neural Network representation, perceptrons, Linear separability and adaline neural net. [10]

Outcome(s): On successful completion of this course, students will be able to
- Solve the basic problems and apply searching techniques used in Artificial Intelligence.
- To select knowledge representations to solve AI problems.
- Apply various machine learning techniques.

Text Books:
1. Rich E., and Knight K., Artificial Intelligence, Tata McGraw-Hill, 3rd edition, 2009.
2. P. Flach, Machine Learning: The art and science of algorithms that make sense of data, Cambridge University Press, 2012.

Reference Books:
1. Patterson D. W., Introduction to Artificial Intelligence and Expert Systems, Prentice-Hall of India, 1st edition, 2015.
2. George F. Luger, Artificial Intelligence: Structures and Strategies for Complex Problem Solving, Pearson Education, 5th edition, 2009.
3. Russell S., and Norvig P., Artificial Intelligence – A Modern Approach, Prentice-Hall of India, 3rd edition, 2014.

CS 23.303 **Operating System** **Total Lectures: 40**
Credit: 3-0-0-3

Prerequisite(s): CS 23.208 (Computer Organization & Architecture)

Objective(s):
- Understand the fundamental operating system abstractions like processes, threads, files, semaphores, IPC and deadlocks.
- Understand operating system from user's perspective as well as from system's perspective.
- Understand the principles of concurrency and synchronization.
- Understand basic resource management techniques.

1. **Overview:** Definition, types of operating systems, functions of operating systemfactors effecting the effectiveness of operating system, multiprocessor,real-time, embedded, system calls and utilities, types of system calls,batch, multiprogramming and multitasking operating system, Interrupts and its mechanism [5]
2. **Process Management:** Process concepts,process state diagram,Process control block,threads, scheduling concepts, types of schedulers, scheduling algorithms, comparison of various CPU scheduling algorithms, multi-level queue scheduling, Inter process communication [8]
3. **Process Synchronization and deadlock:** Independent and concurrent processes, shared memory approach and message passing approach, mutual exclusion, synchronization, inter process communication, critical section problem and its solution to semaphores: binary and counting semaphores, wait/signal operations and their implementation, classical problems of synchronization, deadlock. [11]
4. **Memory Management:** Logical and physical address spaces, base, limit and relocation registers, address binding, dynamic loading and linking, partitioning, swapping, contiguous memory allocation, non-contiguous memory allocation: segmentation, paging and segmentation, Page fault, process to handle page fault, Page replacement algorithms [8]
5. **File Systems:** File concept, user and system programmer view of file systems, different modules of a file system, disk structure, disk-space allocation methods: contiguous, linked, indexed directory structures, file protection, disk scheduling algorithms. [8]

Outcome(s): On successful completion of this course, students will be able to
- Analyze the primary functionalities of operating system.
- Manage system resources efficiently as an operating system designer.
- Develop new operating systems as well as work on existing ones to develop them further.

Text Books:
1. Galvin, Silberschatz, Gagne, Operating System Concepts, Wiley India, 9th edition, 2015.
2. Stallings, Operating Systems: Internals and Design Principles, Pearson education, 7thedition 2013.

Reference Books:
1. Andrew S. Tanenbaum, Modern Operating Systems - Prentice-Hall, Inc., 4th edition, 2015.
2. Deitel, Operating Systems, Pearson Education , 3rd edition, (2012 reprint).

CS 23.304 **Compiler Design** **Total Lectures:40**
Credit: 3-0-0-3

Prerequisite(s): CS 23.206 (Theory of Computation)

Objective(s):

- To introduce the major conceptual areas of language translation and compiler design.
- To enrich the knowledge in various phases of compiler, code optimization techniques
- To extend the knowledge of different parsers.
- To provide practical programming skills necessary for construction of a compiler.

1. **Introduction& Lexical Analysis:** Introduction of compiler, compiler structure: analysis-synthesis model of compilation, various phases and passes of a compiler, Symbol table: Role and data structure used for symbols tables, BootstrappingLexical Analysis: Input buffering, specification & recognition of tokens, error detection &recovery, Finite state machines and regular expressions and their applications to lexical analysis, Optimization of DFA-Based Pattern Matchers implementation of lexical analyzers. [8]
2. **Syntax Analysis:** Syntax analysis: CFGs, top down parsing, Brute force approach, recursive descent parsing, transformation on the grammars, predictive parsing, bottom up parsing, operator precedence parsing, LR parsers, parser generation error detection & recovery. [9]
3. **Syntax Directed Definition& Syntax Directed Translation**: Syntax-directed Translation schemes, Implementation of Syntaxdirected Translators, Construction of syntax trees, bottom up evaluation, S-attribute definition, L-attribute definition, top down translation, bottom up evaluation of inherited attributes, recursive evaluation, and analysis of syntax directed definition. [8]
4. **Intermediate Code Generation:** Intermediate code, postfix notation, parse trees & syntax trees, three address code, quadruple & triples, translation of assignment statements, Boolean expressions, statements that alter the flow of control, postfix translation, translation with a top down parser, more about translation: array references in arithmetic expressions, procedures call, declarations, case statements. [8]
5. **Code Optimization & Code Generation:** Loop optimization, the DAG representation of basic blocks, flow graphs, global data-flow analysis, peephole optimization, generating code from DAG, register allocation and assignment. [7]

Outcome(s): On successful completion of this course, the students will be able to

- Use lexical analyzer and parser generator tools.
- Generate intermediate code.
- Be familiar with register allocation.
- Expose compiler code optimization.

Text Books:

1. A. V. Aho, R. Sethi, and J. D. Ullman. Compilers: Principles, Techniques and Tools, Pearson Education, 2nd edition, 2014.
2. Louden, Compiler Construction: Principles and Practice, Cengage Learning, 1st edition, 1997.

Reference Books:

1. Allen I. Holub, Compiler Design in C, Prentice-Hall Inc., 1st edition, 1994.
2. AW Appel, M Ginsburg, Modern Compiler Implementation in C, Cambridge University Press, 1st edition, 2004.
3. J. R. Levine, T. Mason, D. Brown, Lex and Yacc, O'Reilly, 2nd edition, 2000.

CS 23.306 **Computer Networks** **Total Lectures: 40**
Credit: 3-0-0-3

Prerequisite(s): CS 23.203 (Data Structures)

Objective(s):
- To become familiar with layered communication architectures (OSI and TCP/IP).
- To understand the client/server model and key application layer protocols.
- To construct large-scale computer networks
- To understand fundamentals of computer networks and major protocols.
- To analysis and evaluate major Internet protocols.
- To program simple network applications using socket API
- To learn and adopt newly introduced computer network technologies by themselves.

1. **Introduction:** History and development of computer networks, networks topologies, MANET, categories of networks, layering and protocols (OSI and TCP/IP). [5]
2. **Physical and MAC Layer:** Different types of transmission media, transmission impairments, encoding schemes (NRZ, NRZ-I, NRZ-L, RZ, Manchester, 4B/5B, Differential Manchester etc.), packet and circuit switching. Aloha, CSMA/CA, CSMA/CD protocols, performance evaluation and characteristics. [8]
3. **Data Link Layer:** Framing concept, error detection (parity, CRC), sliding window, stop and wait protocols, stop performance and efficiency. IEEE 802.3, 802.4, 802.5, 802.11, performance evaluation. [10]
4. **Network Layer:** IPv4, IP address classes, classless addressing, IPv6, ARP, RARP, DHCP, ICMP, routing algorithms: distance vector, metrics, RIP, OSPF, BGP, congestion control algorithms, leaky bucket, token bucket. [8]
5. **Transport & Application Layer:** Duties of the transport layer, connection, TCP, TCP services, TCP header, UDP, evaluating protocol such as TCP/UDP using simulator like NS2. DNS, FTP, HTTP, SMTP, POP, basics of network security, cryptography, digital signature and firewall. [9]

Outcome(s): On successful completion of this course, students will be able to
- Analyze computer network component and devices.
- Synthesize how communication works in a network and internet.
- Formulate the concept of topologies and their applications.
- Analyze protocol at different layers and how they play important role in network implementation.

Text Books:
1. Tanenbaum A. S., Computer Networks, Prentice-Hall of India, 5th edition, 2010.
2. Forouzan B. A., Data Communications and Networking, Tata McGraw-Hill, 5th edition, 2013.

Reference Books:
1. Stallings W., Data and Computer Communications, Prentice-Hall of India, 10th edition, 2013.
2. Kurose and Ross, Computer Network: a top-down approach, Pearson 6th edition 2015.

CS 19.307 **Computer Graphics and Multimedia** **Total Lectures: 40**
Credit: 3-0-0-3

Prerequisite(s): MA 19.101 Mathematics-I

Objective(s):

- To understand the basics of various inputs and output computer graphics hardware devices
- To understand an in-depth exploration of fundamental concepts in 2D and 3D computer graphics.
- To study 2D raster graphics techniques, 3D modeling, geometric transformations, 3D viewing and rendering.

1. **Introduction:**Applications of Computer graphics, Graphic Displays: Random scan displays, Raster scan displays, Frame buffer and video controller, Points and lines, Line drawing algorithms, Circle generating algorithms. [8]
2. **Transformations:** Basic transformation, Matrix representations and homogenous coordinates, Composite transformations, Reflections and shearing. [10]
   Windowing and Clipping: Viewing pipeline, Viewing transformations, 2-D Clipping algorithms- Line clipping algorithms such as Cohen Sutherland line clipping algorithm, Liang-Barsky algorithm, Line clipping against non-rectangular clip windows, Polygon clipping: Sutherland Hodgeman polygon clipping, Weiler-Atherton polygon clipping, Curve clipping, Text clipping.
3. **Three Dimensional:** 3-D geometric primitives, 3-D Object representation, 3-D Transformation, 3-D viewing, projections, 3-D Clipping. [6]
4. **Curves and Surfaces:** Quadric surfaces, Spheres, Ellipsoid, Blobby objects, Introductory concepts of Spline, B-Spline and Bezier curves and surfaces. Back Face Detection algorithm, Depth-buffer method, A-buffer method, Scan line method, basic illumination models: Ambient light, 08 Diffuse reflection, Specular reflection and Phong model. [8]
5. **Multimedia:** Overview, applications, multimedia hardware, multimedia authoring tools, audio format, digital video format, compression techniques JPG, MPEG, basic animation concepts. [8]

Outcome(s): On successful completion of this course, the students will be able to

- To understand the various computer graphics hardware and display technologies.
- 2D and 3D viewing technologies
- Various 2D and 3D objects transformation techniques

Text Books:

1. D.Hearn and P. Baker, "Computer Graphics C Version" - Pearson Education –2nd Edision 2002
2. Vaughan T. , Multimedia making it work, TMH 3rd Edition 2002.

Reference Books:

1. Rogers D, Procedural elements for computer graphics, TMH, 2nd Edition 1997.
2. Rogers D, Mathematical elements for computer graphics, TMH, 2nd Edition 2002
3. Mukherjee, Fundamentals of Computer graphics & Multimedia, PHI 3rd Edition 2010.

CS-23.308 **Computer Graphics and Digital Image Processing** **Total Lectures: 40**
Credit: 3-0-0-3

Prerequisite(s): CS-19.101 (Programming Language)

Objective(s):

- To study the image fundamentals and mathematical transforms necessary for Computer graphics
- To study the image enhancement techniques
- To study image restoration procedures.
- To study the image compression procedures.
- To study the image segmentation and representation techniques
- To study the image representation in form of features and descriptors

1. **Basics of Computer Graphics:** Types of computer graphics, Line drawing algorithms, Circle generating algorithms, **Transformations:** Basic transformation, Matrix representations and homogenous coordinates, Composite transformations, Reflections and shearing. Windowing and Clipping: Viewing pipeline, Viewing transformations, 2-D Clipping algorithms- Line clipping algorithms Polygon clipping [9]
2. **Digital Image Fundamentals and Image Enhancement:** Elements of digital image processing systems, elements of visual perception, Image Transformation Techniques: DFT, DCT, KLT, SVD, **Image Enhancement-** Histogram equalization and specification techniques, noise distributions, spatial averaging, directional smoothing, median, geometric mean, harmonic mean, contra harmonic mean filters, image restoration [12]
3. **Morphological Image Processing: S**tructuring element, erosion, dilation, opening, closing, hit-or-miss transform, boundary detection, hole filling, connected components, convex hull, thinning, thickening, skeletons, pruning. [7]
4. **Image Segmentation: T**hresholding , region based segmentation, region growing, region splitting and merging, and segmentation by morphological watersheds, basic concepts, dam construction, and watershed segmentation algorithm, Edge detection, edge linking via Hough transform, [6]
5. **Image Representation:** Boundary representation, chain code, polygonal approximation, signature, boundary segments, boundary description, shape number, Fourier descriptor, regional descriptors, and topological features. [6]

Outcome(s): On successful completion of this course, students will be able to

- To understand the various computer graphics hardware and display technologies
- Analyze the basic algorithms used for image processing &image compression with morphological image processing.
- Apply image enhancement and restoration techniques.
- Use image compression and segmentation Techniques.
- Represent features of images.

Text Books:

1. Rafael C. Gonzales, Richard E. Woods, "Digital Image Processing", Third Edition, Pearson Education, 2010.
2. D.Hearn And P.Baker, Computer Graphics, - Pearson Education
3. Anil Jain K. "Fundamentals of Digital Image Processing", PHI Learning Pvt. Ltd., 2011.

Reference Books:

1. Rafael C. Gonzalez, Richard E. Woods, Steven L. Eddins, "Digital Image Processing Using MATLAB", Third Edition Tata Mc Graw Hill Pvt. Ltd., 2011.
2. William K Pratt, "Introduction to Digital Image Processing", CRC Press, 13-Sep-2013.
3. Bernd Jähne, Digital Image Processing, Springer Science & Business Media, 29-Jun-2013.

CS 23.311 **Data Mining and Warehousing** **Total Lectures: 40**
Credit: 3-0-0-3

Prerequisite(s): **NIL**

Objective(s):

- To understand data warehouse concepts, architecture, business analysis and tools
- To learn data mining and warehousing and knowledge base
- To learn data marts OLAP

1. **Overview:** Concept of data mining and warehousing, data warehouse roles and structures, cost of warehousing data, roots of data mining, approaches to data exploration and data mining, foundations of data mining, web warehousing, web-warehousing for business applications and consumers, introduction to knowledge management, data warehouses and knowledge bases. [8]
2. **Data Warehouse:** Theory of data warehousing, barriers to successful data warehousing, bad data warehousing approaches, stores, warehouse and marts, data warehouse architecture, data about data - metadata, metadata extraction, implementing the data warehouse, data warehouse technologies, Multi Dimensional Space – Constraint Based Frequent Pattern Mining, Classification using Frequent Pattern . [8]
3. **Data Mining and Data Visualization:** Data mining, OLAP, techniques used to mine the data, market basket analysis, current limitations and challenges to DM, data visualization. [8]
4. **Designing and Building the Data Warehouse:** The enterprise model approach of data mining design, data warehouse project plan, analysis and design tools, data warehouse architecture, specification and development. [8]
5. **Web-Based Query and Reporting:** Delivering information over the web, query and reporting tools and business value, architectural approaches to delivering query capabilities over the web. Web Based Statistical Analysis and Data Mining: Analytical tools, business value from analytical tools, [8]

Outcome(s):

- To understand data mining and data warehousing
- To understand and use tools for data mining

Text Books:

[1] Marakas G. M., “Modern Data Warehousing, Mining and Visualisation”, Pearson Education Asia, 2003.
[2] Mattison R., “Web Warehousing and Knowledge Management”, Tata McGraw-Hill,2002.

Reference Books:

[1] Mallach E., “Decision Support and Data Warehousing”, Tata McGraw-Hill,2002.
[2] Hand D., Mannila H., and Smyth P., “Principles of Data Mining”, Prentice-Hall
of India, 2001.
[3] Alex Berson and Stephen J.Smith, ―Data Warehousing, Data Mining & OLAP‖, Tata McGraw – Hill Edition, 35th Reprint 2016.

CS23.305 Cyber Security **Total Lectures: 40**

Credits: 3-0-0-3

Prerequisite(s):
- Understand basic principles of network connectivity.
- Understand basic operating system fundamentals including Windows or Linux.

Objective(s): A graduate of this program will be able to:

- Evaluate specific security techniques used to administer a system that meets industry standards and core controls.
- Explain methods for establishing and maintaining the security of a network, computing environment, and application.
- Apply control techniques to secure networks, operating systems, and applications.
- Conduct threat assessments and vulnerability scans to secure the assets of an organization.

1	**Introduction:** Introduction to Cyber Security - Importance and challenges in Cyber Security - Cyberspace – Cyber threats - Cyber warfare - CIA Triad - Cyber Terrorism - Cyber Security of Critical Infrastructure - Cyber security - Organizational Implications.	[8]
2	**Hackers and Cyber Crimes:** Types of Hackers - Hackers and Crackers - Cyber-Attacks and Vulnerabilities - Malware threats - Sniffing - Gaining Access - Escalating Privileges - Executing Applications - Hiding Files – Covering Tracks - Worms - Trojans - Viruses - Backdoors	[9]
3	**Ethical Hacking and Social Engineering:** Ethical Hacking Concepts and Scopes - Threats and Attack Vectors - Information Assurance – Threat Modeling - Enterprise Information Security Architecture - Vulnerability Assessment and Penetration Testing - Types of Social Engineering - Insider Attack - Preventing Insider Threats – Social Engineering Targets and Defence Strategies.	[8]
4	**Cyber Forensics and Auditing:** Introduction to Cyber Forensics - Computer Equipment and associated storage media - Role of forensics Investigator - Forensics Investigation Process - Collecting Network based Evidence - Writing Computer Forensics Reports - Auditing - Plan an audit against a set of audit criteria - Information Security Management System Management. Introduction to ISO 27001:2013.	[8]
5	**Cyber Ethics and Laws:** Introduction to Cyber Laws - E-Commerce and E-Governance - Certifying Authority and Controller - Offences under IT Act-	[7]

Computer Offences and its penalty under IT Act 2000 - Intellectual Property Rights in Cyberspace.

Outcome(s): At the end of the course the students will be able to do following:

- Understand the basic concepts of cyber security and cyber crimes.
- Understand the security policies and cyber laws.

Text Books:

[1] Nina Godbole, Sumit Belapure, "Cyber Security", Willey, 2011.
[2] Cyber Law By Bare Act, Govt of India, It Act 2000.
[3] James Graham, Ryan Olson, Rick Howard, "Cyber Security Essentials", CRC Press,15-Dec 2010.

Reference Books:

[1] Roger Grimes, " Hacking the Hacker" , Wiley, ist Edition, 2017.
[2] Donaldson, S., Siegel, S., Williams, C.K., Aslam, A., "Enterprise Cyber security -How to Build a Successful Cyber defense Program against Advanced Threats", Apress, 1st Edition, 2015.
[3] SunitBelapure and Nina Godbole, "Cyber Security: Understanding Cyber Crimes, Computer Forensics And Legal Perspectives", Wiley India Pvt Ltd, ISBN: 978-81-265-21791, Publish Date 2013

CS 23.351 **DAA Laboratory** **Total Hrs: 26**
Credit: 0-0-2-1

Prerequisite(s): CS 23.253(Data Structure Laboratory)

Objective(s):

- To analyze the asymptotic performance/Time complexity of algorithms.
- To demonstrate familiarity with major algorithms and data structures.
- To apply important algorithmic design paradigms, techniques and methods for the solution of real life problems.
- To synthesize efficient algorithms in common engineering design situations.

Contents: Lab manual based upon the theory course CS 23.301 and to achieve the listed course objectives and outcomes.

- To implement and analyze sorting algorithms Insertion, Quick and Merge Sort
- To implement greedy algorithms Knapsack problem
- To implement Dikstra and Warshall's Algorithm
- To implement Bellman Ford Algorithm
- To implement Minimum spanning tree algorithms
- To implement DFS and BFS

Outcome(s): On successful completion of this course, the students will be able to

- Understand and analyze the time complexities of sorting algorithms
- Understand and analyze Graph algorithms
- Understand and analyze the DFS and BFS algorithms

CS-23.354 **Compiler Design Laboratory** **Total Hours: 26**
Credit: 0-0-2-1

Prerequisite(s): CS 23.111(Programming Lab.) CS 23.253(Data Structures Lab.)

Objective(s):

- To design and implement a lexical analyzer
- To design and implement a parser
- To design and implement symbol tables
- To understand and design code generation schemes
- To learn the working of some compiler writing tools

Contents:

- To study the Flex environment
- Implementation of Simple Programs using FLEX
- To illustrate the concept of Special Directives in FLEX
- Illustrate the special functions like yymore(), yyless(), unput(), input(), yyterminate(), yyflush buffer
- To implement the Lexical Analyser using FLEX
- To implement syntax analysis programs such as string check, identifier, integer, converting and evaluation of postfix, infix expressions in YACC
- To demonstrate symbol table implementation in YACC and FLEX
- Implementation of intermediate code generation.

Outcome(s): On successful completion of this course, the students will be able to

- Understand the peculiarities of a language translator
- Get the practical approach of how a compiler works
- Work in the development phase of new computer languages in industry
- Understand the working of compiler construction tools
- Visualize and design a new compiler of her own language

CS 23.356 **Computer Network Laboratory** **Credit: 0-0-2-1**

Prerequisite(s): CS 23.111 (Programming Lab. I), CS 23.253(Data Structures Lab.)

Objective(s):

- To implement classification of IPV4, IPV6.
- To identify the classes of IP Address.
- Network protocol Analyzer using Wireshark
- To evaluate IP Header Checksum
- To calculate the parity check and convert it into even or odd parity
- To evaluate Cyclic Redundancy Check
- To evaluate Longitudinal Redundancy Check
- Introduction to socket programming

Contents:

- To implement classification of IPV4, IPV6.
- Network protocol Analyzer using Wireshark
- To evaluate IP Header Checksum
- To calculate the parity check and convert it into even or odd parity
- To evaluate Longitudinal Redundancy Check
- Introduction to socket programming
- To demonstrate sending a UDP packet and TCP

Outcome(s): On successful completion of this course, students will be able

- To know how communication works at different layers of TCP/IP model
- To understand practically the purpose of various key protocol.
- To learn how a network can be efficiently deployed.
- To identify and analyze the latest technology with existing one.
- To configure network with client/server programming
- To understand computer network component and devices.

CS-23.357	Computer Graphics Laboratory	Total Hours:26 Credit: 0-0-2-1

Prerequisite(s): CS 23.253(Data Structure Laboratory)

Objective(s):
To introduce the use of the components of a graphics system and become familiar with building approach of graphics system components and algorithms related with them.
To learn the basic principles of 3- dimensional computer graphics.
Provide an understanding of how to scan convert the basic geometrical primitives, how to transform the shapes to fit them as per the picture definition.

Contents:
Lab manual based upon the theory course CS 23.301 and to achieve the listed course objectives and outcomes.
Study of basic graphics functions defined in “graphics.h”.
Write a program to draw a hut or another geometrical figures.
Write a program to draw a line through Bresenham’s Algorithm.
Write a program to draw a line using DDA algorithm.
Write a program to draw a line using Mid-Point algorithm.
Write a program to draw a circle using mid-point algorithm.
Write a program to rotate a Circle around any arbitrary point or around the boundary of another circle.
Write a menu driven program to rotate, scale and translate a line point, square, triangle about the origin.
Write a program to perform shearing on a line.
Write a program to implement transformations in three dimensions.
Write a program to implement Liang Barksy Line Clipping Algorithm.
Write a program to implement reflection of a point, line.
Write a program to implement polygon filling.
To implement Cohen–Sutherland 2D clipping and window–viewport mapping.

Outcome(s):
On successful completion of this course, the students will be able to
To implement various algorithms to scan, convert the basic geometrical primitives, transformations, Area filling, clipping.
To describe the importance of viewing and projections.
To define the fundamentals of animation, virtual reality and its related technologies.

CS-23.358 **Computer Graphics and Digital Image Processing Laboratory** **Total Hours: 26**

Credit: 0-0-2-1

Prerequisite(s): CS 19.151 (Programming Language Laboratory.)

Objective(s): The aim of this course is to familiarize students with the implementation of

- algorithms to draw basic geometric objects and their transformations in Computer Graphics
- basic techniques for handling images and developing algorithms for image processing (quality enhancement, restoration, coding, compression).

Contents:

- Implementation of Scan conversion algorithms
- Introduction to MATLAB
- Implementation of Image Transformation Fourier, DCT etc.
- Implementation of Image Enhancement operations.
- Implementation of image degradation and restoration methods
- Implementation of image segmentation and morphological operations.

Outcome(s): On successful completion of this course, students will be able to

- Implement scan conversion algorithms of basic geometrical primitives
- Learn basic image processing techniques for solving real problems.
- Learn two-dimensional signal processing and its wide range of applications, for example, image restoration, image compression, and image analysis.

CS23.355 **Cyber Security Lab** **Total Hours: 26**

Credits: 0-0-2-1

Prerequisite(s): A course on "Network Security and Cryptography".

Objective(s): To get practical exposure of Cyber security threats and Forensics tools.

List of Experiments

1 Perform an Experiment for port scanning with nmap
2 Set Up a honeypot and monitor the honeypot on the network
3 Install Jscript/Cryptool tool (or any other equivalent) and demonstrate Asymmetric, Symmetric crypto algorithm, Hash and Digital/PKI signatures.
4 Generate minimum 10 passwords of length 12 characters using open SSL command
5 Perform practical approach to implement Footprinting-Gathering target information using Dmitry-Dmagic, UAtester
6 Working with sniffers for monitoring network communication (Wireshark).
7 Using Snort, perform real time traffic analysis and packet logging.
8 Perform email analysis using the Autopsy tool.
9 Perform Registry analysis and get boot time logging using process monitor tool
10 Perform File type detection using Autopsy tool
11 Perform Memory capture and analysis using FTK imager tool
12 Perform Network analysis using the Network Miner tool

Outcome(s): At the end of the course the students will be able to do following:

- Get the skill to identify cyber threats/attacks.
- Get the knowledge to solve security issues in day to day life.
- Able to use Autopsy tools
- Perform Memory capture and analysis
- Demonstrate Network analysis using Network miner tools

Text Books:

[1] Real Digital Forensics for Handheld Devices, E. P. Dorothy, Auerback Publications, 2013.
[2] The Basics of Digital Forensics: The Primer for Getting Started in Digital Forensics, J. Sammons, Syngress Publishing, 2012.

Reference Books:

[1] Handbook of Digital Forensics and Investigation, E. Casey, Academic Press, 2010.
[2] Malware Forensics Field Guide for Windows Systems: Digital Forensics Field Guides, C. H. Malin, E. Casey and J. M. Aquilina, Syngress, 2012.
[3] The Best Damn Cybercrime and Digital Forensics Book Period, J. Wiles and A. Reyes, Syngress, 2007.

CS 23.310 **Soft Computing** **Total Lectures: 40**
Credit: 3-0-0-3

Prerequisite(s): (Logic in Computer Science)

Objective(s):

- To familiarize with soft computing concepts
- To introduce the idea of neural networks, fuzzy logic and use of heuristics based on human experience
- To introduce the concept of genetic algorithm and its applications to soft computing using some applications

1. **Introduction**: Introduction of soft computing, types of soft computing, applications of soft computing, introduction of artificial intelligence, production system, various search techniques, hill climbing, best first search, knowledge representation, monotonic and non-monotonic reasoning. [8]
2. **Neural Networks:**Structure and function of a single neuron, definition of ANN, characteristics of ANN, Terminologies of ANN, single layer networks, perceptron, Concept of linear separability, Hebb's Network, Supervised learning networks: perceptron learning, delta rule, MADALINE, Activation functions, Back propagation algorithms, characteristics of EBPA. [8]
3. **Neural Network Architectures**: Counter propagation networks, architecture, functioning, characteristics, Hopfield network, configuration, associative memory, limitations and applications, Hopfield vs Boltzman machine, adaptive resonance theory, architecture, classification, implementation and training [8]
4. **Fuzzy Logic :**Fuzzy logic, fuzzy set theory, fuzzy set vs crisp set, crisp relation and fuzzy relation, introduction and features of membership functions, fuzzy rule based system, fuzzy propositions, formation, decomposition and aggregation of fuzzy rules, fuzzy reasoning, fuzzy inference system, fuzzy decision making, applications of fuzzy logic. [8]
5. **Genetic Algorithms :**Genetic algorithm, fundamentals, basic concepts, working principle, encoding, fitness function, reproduction, genetic modeling, inheritance operator, cross over, inversion and deletion, mutation operator, bitwise operator, generational cycle, convergence of GA, applications and advances in GA, difference and similarities between GA and other traditional methods. [8]

Outcome(s): On successful completion the students will be able to:

- Identify and describe soft computing techniques and their roles in building intelligent machines
- Apply fuzzy logic and reasoning to handle uncertainty and solve engineering problems
- Apply genetic algorithms to combinatorial optimization problems
- Apply neural networks to pattern classification and regression problems
- Evaluate and compare solutions by various soft computing approaches for a given problem

Text Books:

1. S. N. Sivanandan and S.N. Deepa, "Principles of Soft Computing", Wiley India, 2007
2. J.S.R.Jang, C.T.Sun and E. Mizutani, "Neuro-Fuzzy and Soft Computing", PHI, 2004
3. Pravir K. Chawdhry, Rajkumar Roy, Raj K. Pant "Soft Computing in Engineering Design and Manufacturing"Springer, 2012.

Reference Books:

1. S. Rajasekaran and G.A.V. Pai, "Neural Networks, Fuzzy Logic and Genetic Algorithms", PHI, 2003
2. Timothy J.Ross, "Fuzzy Logic with Engineering Applications", McGraw-Hill, 3rd Edition, 2009.
3. Davis E. Goldberg, "Genetic Algorithms: Search, Optimization and Machine Learning", Pearson Education , 2006

CS 23.391 **Blockchain Components and Architecture** **Total Lectures: 40**
Credit: 3-0-0-3

Prerequisite(s): Basic concepts in blockchain and networking.

Objective(s):

- To described implementation and architecture of blockchain
- To implement Blockchain in Financial Software and Systems.
- To introduce the Blockchain history, Digital Money, Hash, Signature and Blockchains design goals.

1. Introduction to Blockchain history: Digital Money to Distributed Ledgers Design Primitives: Protocols, Security, Consensus, Permissions, Privacy, Blockchain Architecture and Design. Basic crypto primitives: Hash, Signature, Hashchain to Blockchain, Basic consensus mechanisms [8]
2. Consensus, Requirements for the consensus protocols, Proof of Work (PoW) Scalability aspects of Blockchain consensus protocols [5]
3. Permissioned Blockchains, Design goals, Consensus protocols for Permissioned Blockchains Hyperledger, Decomposing the consensus process Hyperledger fabric components Chaincode Design and Implementation Hyperledger Fabric beyond Chain code fabric SDK and Front End, Hyperledger composer tool [10]
4. Blockchain in Financial Software and Systems (FSS):Settlements, KYC, Capital markets, Insurance Use case II: Blockchain in trade supply chain: Provenance of goods, visibility, tradesupply chain finance, invoice management discounting, etc [11]
5. Blockchain Cryptography Privacy and Security on Blockchain, Blockchain consensus protocols, Various recent works on scalability [6]

Outcome(s): On the successful completion of the course, the student will be able to:

- To described implementation and architecture of blockchain
- To design & implement blockchain as a digital asset properly.
- To become an efficient blockchain developer.

Text Books: Blockchain Technology Explained, by Alan T. Norman

Reference Books:

1. Blockchain: Ultimate guide to understanding blockchain, bitcoin, cryptocurrencies, smart contracts and the future of money.
2. The Bitcoin Standard: The Decentralized Alternative to Central Banking by Saifedean Ammous

CS 23.463 **Agile Software Development** **Total Lectures: 40**
Credit: 3-0-0-3

Prerequisite(s): CS 23.210 (Software Engineering)

Objective(s):

- To learn the fundamental principles and practices associated with each of the agile development methods.
- To learn how agile methods scale to large and distributed projects.
- To learn the essentials of collaboration as they apply to agile methods.
- To perform in-depth explorations into aspects of agile development that are particularly relevant to each student through detailed discussion sessions.

1	**Fundamentals of Agile:** The genesis of Agile, introduction and background, Agile manifesto and principles, overview of Scrum, extreme programming, feature driven development, lean software development, Agile project management, design and development practices in Agile projects, test driven development, continuous integration, refactoring, pair programming, simple design, user stories, Agile testing, Agile tools.	[7]
2	**Agile Scrum Framework:** Introduction to Scrum, project phases, Agile estimation, planning game, product backlog, sprint backlog, iteration planning, user story definition, characteristics and content of user stories, acceptance tests and verifying stories, project velocity, burn down chart, sprint planning and retrospective, daily scrum, Scrum roles – product owner, Scrum master, Scrum team, Scrum case study, tools for Agile project management.	[9]
3	**Agile Testing:** The Agile lifecycle and its impact on testing, test driven development (TDD), unit framework and tools for TDD, testing user stories – acceptance tests and scenarios, planning and managing testing cycle, exploratory testing, risk-based testing, regression tests, test automation, tools to support the Agile tester, securing Agile software.	[10]
4	**Agile Software Design and Development:** Agile design practices, role of design principles including single responsibility principle, open closed principle, Liskov substitution principle, interface segregation principles, dependency inversion principle, Agile design, need and significance of refactoring, refactoring techniques, continuous integration, automated build tools, version control.	[10]
5	**Industry Trends:** Market scenario and adoption of Agile, Agile ALM, roles in an Agile project, Agile applicability, Agile in distributed teams, business benefits, challenges in Agile, risks and mitigation, Agile project on cloud, balancing Agility with discipline, Agile rapid development technologies.	[4]

Outcome(s): On successful completion of this course, students will be able to

- Understand concept of agile software engineering and its advantages in software development. Recognize various agile methods.
- Design and manage a project using Scrum.
- Write user stories for project requirements.
- Write test cases for developing features.

Text Books:

1. Ken Schawber, Mike Beedle, Agile Software Development with Scrum, Pearson, 2008.
2. Robert C. Martin, Agile Software Development Principles, Patterns and Practices, Prentice Hall

CS 23.412 **Big Data and Unstructured Databases** **Total Lectures: 40**
Credit: 3-0-0-3

Prerequisite(s): CS 19.204 (Database Management System)

Objective(s): To understand fundamental concept of Big Data and its application in various fields

1	**Introduction to Big Data**: Big data vs RDBMS, characteristics of big data, semi-structured data, structured data, sources of data in business context, data analytics and statistics, advantages of big data analytics. use of big data in preventing fraudulent activities.	[5]
2	**Technologies for Handling Big Data**: Distributed and parallel computing for big data, massively parallel processing,cloud computing and big data, introduction to Hadoop ecosystem, functioning of hadoop, HDFS, job tracker, task tracker, name node, secondary name node, data node, Hadoop 1.0 and Hadoop 2.0, YARN, In-memory computing, challenges with big data, integrity and privacy of big data.	[9]
3	**Big Data Stack and Analytics Architecture:** Descriptive, diagnostic, discovery, prescriptive and predictive analytics, , load leveling with queues, load balancing with multiple consumers, leader election, sharding, CAP theorem, bloom filter, scheduler-agent-supervisor, pipes and filters, consensus in distributed systems, map reduce patterns.	[8]
4	**Big Data Analytics Implementations:** Data acquisition, publish-subscribe messaging frameworks, big data collection systems, messaging queues, custom connectors, big data storage, batch analysis, real time analysis, interactive querying, serving databases and web frameworks, data visualization.	[10]
5	**NoSQL Data Management**: introduction to NoSQL, aggregate data models, key value data model, document databases, relationships, graph databases, schema-less databases, map reduce partitioning and combining, composing mapreduce calculations.	[8]

Outcome(s): On successful completion of this course, students will be able to

- Apply knowledge of Big Data to solve real life problems in various areas of social importance

Text Books:

1. Big Data, Black Book: Covers Hadoop 2, Mapreduce, Hive, Yarn, Pig, R And Data Visualization, Dreamtech press, 2016
2. Seema Acharya, Subhashini Chellapan, “Big Data and Analytics”, Wiley, ISBN:978-81-265-5478-2, 2015
3. A Bagha, V Madisetti, "Big Data Science and Analytics: A Hands-On Approach", Arshdeep Bahga & Vijay Madisetti, 2016

Reference Books:

1. Rick Smolan, Jennifer Erwitt, “The Human Face of Big Data”, Sterling, 2012, ISBN:9781454908272
2. EMC Education Services, “Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data”, John Wiley & Sons, 05-Jan-2015, ISBN: 978-1-118-87613-8
3. Tom White, “Hadoop: The Definitive Guide”, O’Reilly Media, 2012, ISBN: 978-1-449-38973-4

CS 23.431 **Advanced Java** **Total Lectures: 40**
Credit: 3-0-0-3

Prerequisite(s): CS 23.202 (Java Programming)

Objective(s): This course extends the study of Web Programming Fundamentals. Advanced concepts of program design, implementation and testing will be introduced with J2EE framework. Student will learn various concept of advance Java for implementation of small to medium sized application programs that illustrate professionally acceptable coding and performance standards.

1	**Introduction:** OOPS principle, class and objects, basics of HTML, web development introduction, web framework, J2EE building blocks, web application structure	[4]
2	**Servlets:** Servlet needs, servlet request-response model, servlet life cycle (init, service, doGet, doPost), servlet configuration, scopes (session, request, page), servlet parameters, servlet context, forms processing (get, post), HTTP header processing, HTTP responses.	[10]
3	**Java Server Pages**: JSP life cycle, JSP a servlet, creating dynamic content with use of JSP elements: expressions, scriptlets& declarations, JSP composition, redirecting, forwarding, accessing default objects (request, response, session, application, config, pageContext), error processing with use of JSP, Servlets vs. JSP pages, Connection to SQL and No SQL databases, JSP interaction with JSON	[10]
4	**Session management**: Http session, the Http session interface session tracking: cookies, hidden fields, cookies and JSP, sending and receiving cookies, Designing of Shopping basket.	[6]
5	**Using Beans and tag libraries**: Separating presentation and business logic, JSP actions, JavaBeans, working with properties, <jsp:useBean>, <jsp:getProperty> and <jsp:setProperty>,using form parameters with Beans, custom tags, types of tags.	[10]

Outcome(s): Upon successful completion of this course you should be able to:

- Demonstrate knowledge of the advance Java along with object oriented development process and apply this understanding to the analysis and design of solutions for small to medium scale problems.
- Implement web application using all the feature of J2EE covered in the course.

Text Books:

1. Phillip Hanna, Osborne, “JSP 2.0: The Complete Reference”, 2nd ed, McGraw-Hill,2003.
2. Martin Bond, Debbie Law, Andy Longshaw, Dan Haywood, Peter Roxburgh, “SAMS Teach Yourself J2EE in 21 Days”, 2nd ed, SAMS, 2004.

Reference Books:

1. Bryan Basham, Kathy Sierra & Bert Bets, Head First Servlets and JSP, 2nd Edition, , Oreilly publication, 2008.

CS 23.441 **Shell Programming** **Total Lectures: 40**
Credit: 3-0-0-3

Prerequisite(s): CS 23.101 (Programming fpr Problem Solving-I)

Objective(s):
- To familiarize students with the Unix/Linux environment
- To learn the fundamentals of shell scripting/programming
- To familiarize students with basic Unix/Linux administration

1	**UNIX-Basic Concepts:** Introduction, file systems, visual editors, Unix philosophy and features, understanding the Unix commands and general-purpose utilities and navigating the file system, the shell and kernel.	[8]
2	**UNIX-Advanced Features:** File and directory structures, file systems and peripheral devices, handling ordinary files, file attributes. UNIX system administration, data blocks and i-nodes.	[8]
3	**Unix Editors and Filters:** Vi Editor, simple filters, regular expression and grep family, egrep, advanced filters: sed and awk, awk pattern scanning and processing, awk arithmetic operators and variables.	[8]
4	**Shell Fundamentals:** Command syntax, Shell variables, pattern matching, escaping, command substitution, input /output redirection, pipes, tees.	[6]
5	**Shell Scripting:** Bourne shell, C shell; scripts, meta characters and environment, shell programming and scripts, logical operators, control and looping statements and programs. Shell reliability, maintainability, efficiency, portability and usability.	[10]

Outcome(s): After completion of the course students will be able to
- Work confidently in Unix/Linux environment
- Write shell scripts to automate various tasks
- Master the basics of Unix/Linux administration
- Understand the basic difference between high level programming and shell programming

Text Books:
1. Sumitabha Das, "UNIX", Tata McGraw-Hill, 4th ed, 2011.
2. Cameron Newham, "Learning the bash Shell: Unix Shell Programming", 3rd ed, O'Reilly Media, 2009.

Reference Books:
1. Bill Rosenblatt, "Learning the Korn Shell", O'Reilly Media, 2002.
2. Mark G. Sobell, *"A practical guide to Linux* commands, editors, and shell programming", 2nd ed, Prentice Hall, 2013.

CS 23.461 **Cloud Computing** **Total Lectures: 40**
Credit: 3-0-0-3

Prerequisite(s): CS 23.306 (Computer Networks)

Objective(s): To learn how to use cloud services.
To implement virtualization and task scheduling

1	**Overview of Cloud Computing:** Brief history and evolution - history of cloud computing, evolution of cloud computing, traditional vs. cloud computing, cloud service models (IaaS, PaaS & SaaS), cloud deployment models (public, private, hybrid and community cloud), benefits and challenges of cloud computing, introduction to AWS public cloud vendor, cost optimization in AWS, basics of virtualization, virtualization technologies, server virtualization, VM migration techniques, role of virtualization in cloud computing, introduction to EC2 service of AWS.	[10]
2	**Working with Private Cloud:** Private cloud definition, characteristics of private cloud, private cloud deployment models, private cloud vendors - CloudStack, OpenStack, Eucalyptus Microsoft, private cloud – benefits and challenges, private cloud implementation in Amazon EC2 service.	[8]
3	**Working with Public Clouds:** What is public cloud, why public cloud, when to opt for public cloud, public cloud service models, public cloud players, infrastructure as a service offering, IaaS vendors, PaaS offerings, PaaS vendors, software as a service, demonstrating public cloud with AWS – storage and database services, private vs. public cloud – when to choose.	[10]
4	**Overview of Cloud Security:** Explain the security concerns in traditional IT, introduce challenges in cloud computing in terms of application security, server security, network security, security reference model, abuse and nefarious use of cloud computing, insecure interfaces and APIs, malicious insiders, shared technology issues, data loss or leakage, account or service hijacking, unknown risk profile, shared security model between vendor and customer in IaaS/PaaS/SaaS, implementing security in AWS.	[8]
5	**Future directions in Cloud Computing:** When and not to migrate to cloud, migration paths for cloud, selection criteria for cloud deployment, issues/risks in cloud computing, future technology trends in cloud computing.	[4]

Outcome(s): On successful completion the students will be able to:

- Analyze the cloud computing setup with its vulnerabilities and applications using different architectures.
- Design different workflows according to requirements and apply map reduce programming model.
- Create combinatorial auctions for cloud resources and design scheduling algorithms for computing clouds.

Text Books:

1. Raj Kumar Buyya, James Broberg and Andrezi M Goscinski, “Cloud Computing: Principles and Paradigms”, John Wiley & Sons, 2010
2. Nick Antonopoulos and Lee Gillam, “Cloud computing: Principles, Systems and Applications”, Springer, 2010.

Reference Books:

1. Anthony T Velte, Toby J Velte and Robert Elsenpeter, “Cloud Computing: A Practical Approach”, McGraw Hill, 2010
2. BorkoFurht and Armando Escalante (Editors), “Handbook of Cloud Computing”, Springer, 2010
3. Judith Hurwitz, Robin Bllor, Marcia Kaufman and Fern Halper, “Cloud Computing for Dummies”, John Wiley & Sons, 2010.

CS 23.462 **Mobile App Development** **Total Lectures: 40**
Credit: 3-0-0-3

Prerequisite(s): CS 23.202 (Java Programming)

Objective(s):

- To learn how mobile apps are different from other traditional apps by learning those aspects of mobile programming that make it unique from other platforms.
- Program mobile applications for the Android operating system that use basic and advanced phone features.
- Deploy applications to the Android marketplace for distribution.

1	**Getting started with Mobility:** Mobility landscape, mobile platform, mobile apps development, overview of android platform, setting up the mobile app development environment along with an emulator, a case study on mobile app development.	[8]
2	**Introduction to Kotlin:** Basics of Kotlin, looping jumping branching, I/O, Classes, and Object with other features of OOPs.	
3	**Building blocks of mobile apps: Apps user:** Interface designing-mobile UI resources (Layout, UI elements, Drawable, Menu), activity-states and life-cycle, interaction amongst activities. app-functionality beyond user interface-threads, async task, service –state and life cycle, notification, broad cast receivers, telephony and SMS APIs native data handling-on device file I/O, shared preferences, mobile database such as SQLite, and enterprise data access (via Internet/Intranet)	[14]
4	**Sprucing up mobile apps:** Graphics and animation- custom views, canvas, animation APIs, multimedia-audio/video playback and record, location awareness, and native hardware access (sensor such as accelerometer and gyroscope).	[8]
5	**Testing mobile apps:** Debugging mobile apps, white box testing, black box testing, and test automation of mobile app, JUnit for Android, Robotium, Monkey Talk, Taking apps to market, packaging and distributing.	[6]

Outcome(s): On successful completion of this course, students will be able to

- Understand the android operating system in details
- Create mobile application for various screens
- Deploy on play store market
- Use various sensors available in mobile

Text Books:

1. Barry Burd, Android Application Development All in One for Dummies, Johen Wiley & Sons Inc., 2011.
2. Anubhav Pradhan, Anil V Deshpandy, Mobile App Development, Edition I

Reference Books:

1. Lauren Darcey, Shane Conder, “Teach Yourself Android Application Development in 24 Hours”, SAMS Publishing, 2012, 2ndEdition , 2012.

CS 23.465 **Machine Learning using R** **Total Lectures: 40**

Credit: 3-0-0-3

Prerequisite(s): Probability, Statistics, Introduction to computer programming, Linear Algebra

Objective(s): Understand and Appreciate why data science is gaining importance in today's business world.
Comprehend where data science can be applied in different scenarios across industry domains.
Understand the fundamentals of R language and its usage for statistical computing
Identify various data importing, manipulation, visualization techniques in R and perform exploratory data analysis
Recognize various machine learning techniques such as classification, regression etc. across several use cases

1. **Unit I: Introduction to Data Science and Machine Learning Digital Data:** Structured, Unstructured, Semi-structured data, what is Machine Learning? Why Machine Learning? Concept of Learning, Types of Machine Learning: Supervised Machine Learning, Unsupervised Machine Learning, Semi-supervised Machine Learning, Reinforcement Machine Learning, Industrial applications of Machine Learning across domains such as Healthcare, Finance, Retail etc. [8]
2. **Unit II: R Objects:** CRAN introduction, Data Handling, Introduction to R, why R? Object, Vector, List, Factor, Matrix, Array, Data Frame, Manipulating Objects, Input/Output, R constructs, R Data Analytics packages. [6]
3. **UNIT III: Descriptive Statistics Central tendency and visualisation:** RDispersion – variance, standard deviation, shape – skewness, kurtosis, percentiles, five point summary, boxplots, histograms, barplot, pie chart, scatter plot, two way tables, covariance, correlation, Chi-Square test for two way tables. [8]
4. **Unit IV: Unsupervised Learning –** Clustering, what is Clustering? Applications of Clustering, Similarity measures, Partition based Clustering Techniques – K means clustering, k-medoid, Hierarchical clustering, Density based clustering, Cluster validation. [8]
5. **Unit V: Supervised Learning & Neural Networks:** Regression, Classification, What is Regression? Simple Linear Regression, Multiple Linear Regression, What is Classification? Logistic Regression, Decision Tree, k-Nearest Neighbors, Support Vector Machine, Neural Networks**,** Activation functions, learning rate, Basics of Deep Learning Networks**.** [12]

Outcome(s): At the end of this course, students will be able to:
1. Explain why, where, how and what Machine Learning is.
2. Recognize various Machine Learning Techniques.
3. Apply Machine Learning techniques such as classification, regression.
4. Explore advanced Machine Learning techniques like Support Vector Machines and Deep Learning.

Hardware / Software Requirements

Pentium P5, 3 GHz or higher
8 GB (or higher) RAM, 100 GB (or higher) HD
Windows 8 or 10 (or higher)
R and R Studio

Books:

1. Nina Zumel, John Mount, Manning Shelter Island, "Practical Data Science with R", Second Edition, Manning Publications, 2019.
2. J.Han, M Kamber, J Pei, "Data Mining Concepts and Techniques", 3rd Edition, Morgan Kaufmann, 2011.
3. Pang-Ning Tan, Steinberg, Vipin Kumar, "Introduction to Data Mining", 2nd Edition, Pearson, 2019.
4. Gareth James, Daniela Witten,Trevor Hastie, Robert Tibshirani, "Introduction to Statistical Learning using R", 1st Edition, Springer, New York, NY, 2017.
5. Max Kuhn, Kjell Johnson, "Applied Predictive Modeling", 2nd Edition, Springer; 2018.

CS 23.493	Foundations and Applications of IoT	Total Lectures: 40 Credit: 3-0-0-3

Prerequisite(s): CS 23.101 (Programming for Problem Solving), CS 23.303 (Operating System), CS 23.306 (Computer Networks)

Objective(s):

- To introduce the vision and impact of IoT.
- To understand IoT layered architecture and protocols
- To introduce hardware used to implement IoT applications.
- To understand the challenges that must be overcome by IoT technology.

1	**Introduction to Internet of Things:**Overview and Motivations, Evolution of IoT, Characteristics of IoT, IoT Layered Architecture, General Observations, ITU-T Views**,** Sensing, Actuation, Basics of Networking, Communication Protocols, Sensor Networks.	[6]
2	**RFID:** Introduction, Principle of RFID, Components of an RFID system **EPCGlobal Architecture Framework:** EPCIS & ONS, Design issues, Technological challenges, Security challenges, IP for IoT, Web of Things **Wireless Sensor Networks:** History and context, WSN Architecture, the node, Connecting nodes, Networking Nodes, Securing Communication, WSN specific IoT applications, challenges: Security, QoS, Configuration, Various integrationapproaches.	[10]
3	**Machine-to-Machine Communications,** Difference between IoT and M2M, Interoperability in IoT,IoT Open source architecture (OIC)- OIC Architecture &Design principles- IoT Devices and deployment models. **Interfacing with Arduino:** Introduction to Arduino Programming, Integration of Sensors and Actuators with Arduino	[8]
4	**Working with Raspberry Pi:**Introduction to Python programming, Introduction to Raspberry Pi, Interfacing Raspberry Piwith basic peripherals, Implementation of IoT with Raspberry Pi, Implementation of IoT with Raspberry Pi, SDN for IoT, Data Handling and Analytics	[8]
5	**IoT Applications:**IoT applications for industry: Future Factory Concepts, Cloud Computing, Sensor-Cloud, Smart Cities and Smart Homes, Connected Vehicles, Smart Grid, Industrial IoT, Case Study: Agriculture, Healthcare, Activity Monitoring**.**	[8]

Outcome(s): On successful completion of this course, students will be able to

- Understand the vision of IoT from a global context.
- Understand the fundamental of IoT hardware.
- Understand the use of Devices, Gateways and Data Management in IoT.
- Build architecture in IoT.
- Analyze, design and develop parts of IoT solution.

Text Books:

1. Daniel Minoli, “Building the Internet of Things with IPv6 and MIPv6: The Evolving World of M2MCommunications”, ISBN: 978-1-118-47347-4, Willy Publications
2. Bernd Scholz-Reiter, Florian Michahelles, “Architecting the Internet of Things”, ISBN 978-3-642-19156-5 e-ISBN 978-3-642-19157-2, Springer

Reference Books:

1. Parikshit N. Mahalle& Poonam N. Railkar, “Identity Management for Internet of Things”, River Publishers, ISBN: 978-87-93102-90-3 (Hard Copy), 978-87-93102-91-0 (ebook).
2. Vijay Madisetti and ArshdeepBahga, “Internet of Things (A Hands-on-

Approach)", 1st Edition, VPT, 2014.
3. Peter Waher, "Learning Internet of Things", Packt Publishing Ltd, 2015.
4. Francis daCosta, "Rethinking the Internet of Things: A Scalable Approach to Connecting Everything", 1st Edition, Apress Publications, 2013
5. Jan Holler, VlasiosTsiatsis, Catherine Mulligan, Stefan Avesand, Stamatis Karnouskos, David Boyle, "From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence", 1st Edition, Academic Press, 2014.
6. Malcolm Frank, Paul Roehrig, Ben Pring,"What to Do When Machines Do Everything", John Wiley & Sons, 2017

CS 23.495 **Principles and Practices of Secure Computing** **Total Lectures: 40**
Credit: 3-0-0-3

Prerequisite(s): (Computer Networks),

(Cryptography and Information Security)

Objective(s):

- To understand the technical realization and security of data and computers.
- To understand the industry security standards, regulatory mandates, audit policies and compliance requirements for cloud-based infrastructures.

1	**Authentication, Access Control and Cryptography:** Identification vs. authentication, authentication based on phrases, biometrics and tokens, multifactor authentication, access control, access policies, role-based access control, problems addressed by encryption, trust and certification, digital signatures.	[8]
2	**Programs and Programming:** Unintentional programming oversights, malicious code – malware, countermeasures, browser attacks, web attacks targeting users, obtaining user or website data, email attacks.	[8]
3	**Operating Systems and Network Security:** Security in operating systems, security in design of operating system, rootkit, threats to network communications, wireless network security, denial of service, distributed denial of service, cryptography in network security, network management.	[10]
4	**Database Security:** Introduction to databases, security requirements of databases, reliability and integrity, database disclosure, data mining and big data.	[6]
5	**Cloud Security:** Cloud computing concepts, risk analysis in cloud computing, cloud security tools and techniques, cloud identity management, securing IaaS, privacy principles and policies.	[8]

Outcome(s): On successful completion the students will be able to:

- Define terms related to computer, data and network security.
- Analyze the offered system and point to the potential safety problems.
- Choose and appropriate engineering approach to solving the security problems in various domains.

Text Books:

1. Charles P Pfleeger, Shari Lawrence Pfleeger and Jonathan Margulies, “Security in Computing”, 5th edition, Prentice Hall, 2015.
2. Tim Mather, SubraKumaraswamy and ShahedLatif, “Cloud Security and Privacy” An Enterprise Perspective on Risk and Compliance”, O’Reilly, 2009

Reference Books:

1. David Salomon, “Foundations of Computer Security”, Springer, 2006
2. Chuck Easttom, “Computer Security Fundamentals”, 2nd edition, Pearson Education, 2012

CS 23.494 **Software Defined Networking** **Total Lectures: 40**

Credit: 3-0-0-3

Prerequisite(s): CS 23.210 (Software Engineering), (Computer Networks)

Objective(s):
To explore the basics of SDN
To discuss various types plane
To list out various controllers
To program a simple SDN

1	**Control and Data Planes**: Introduction, evolution vs revolution, the control plane, data plane, moving information between planes, distributed control planes, convergence time, load balancing, creating the MPLS overlay, centralized control planes, logical vs literal, ATM.LANE, route servers.	[8]
2	**SDN Controllers:** Wire protocol, replication, FAWG, hybrid approaches, SDN controllers, VMware, Nicira, OpenFlow, Mininet, NOX/POX, Trema, Ryu, L3VPN, RSPV-TE problem statement, bin-packing, deadlock, Plexxi, Cisco OnePK.	[8]
3	**Network Programmability:** The management interface, the application-network divide, NETCONF and NETMOD, SNMP, modern programmatic interface, XMPP, JSON, I2RS, OpenStack, CloudStack, Puppet, virtual multitenant data center, SDN solution for data center networks.	[8]
4	**Network Function Virtualization**: Virtualization and data plane I/O, services engineered path, service locations and chaining, NFV at ETSI, non-ETSI NFV work, network topology, LLDP, BGP-TE/LS, ALTO, I2RS topology.	[8]
5	**Building an SDN framework**: Build code first, ask questions later, the Juniper SDN framework, IETF SDN framework, open daylight controller/framework, use cases for bandwidth scheduling, manipulation, data center overlays, big data, NFV, and traffic monitoring.	[8]

Outcome(s):
On successful completion the students will be able to:
Describe the key benefits of SDN
Describe the data plane
Configure an SDN friendly network emulator
Program a simple SDN for a given task using a low-level approach and high-level approach
Evaluate a sample SDNs performance and reliability

Text Books:
Ken Gray, Thomas D Nadeau, "SDN: Software Defined Networks", O'Reilly, 2013.

Reference Books:
- Paul Goransson, Chuck Black, "Software Defined Networks: A Comprehensive Approach", Morgan Kaufmann, First edition, 2014.
- Vishal Shukla, "Introduction to Software Defined Networking – OpenFlow&VxLAN", Createspace, 2013
- Patricia A Morreale, James M. Anderson, "Software Defined Networking: Design and Deployment", CRC Press, 2014

CS 23.413 **Distributed Databases** **Total Lectures: 40**

Credit: 3-0-0-3

Prerequisite(s): CS 23.204 (Database Management System)

Objective(s):

- To build on the previous background of database systems by deepening the understanding of the theoretical and practical aspects of the database technologies, showing the need for distributed database technology.
- To introduce the concept and techniques of distributed database including principles, architectures, design, implementation and major domain of application.

1	**Overview:** Features of distributed versus centralized databases, levels of distributed transparency, types of data fragmentation, distributed database access primitives, integrity constraints in distributed databases.	[6]
2	**Distributed Database Design:** A framework for distributed database design, the design of database fragmentation, allocation of fragments, translation of global queries to fragment queries, parametric queries, a framework for query optimization.	[10]
3	**Management of Distributed Transactions:** A framework for transaction management, supporting atomicity of distributed transactions, concurrency control for distributed transactions, architectural aspects of distributed transactions.	[8]
4	**Concurrency Control:** Foundations of distributed concurrency control, distributed deadlocks, concurrency control based on timestamps, basic concepts of reliability, reliability and concurrency control, determining a constant view of the network, checkpoints and cold restart.	[8]
5	**Distributed Database Administration:** Catalog management in distributed databases, authorization and protection, SDD-1 a system for distributed databases, the R* project	[8]

Outcome(s): On successful completion of this course, students will be able to
Identify the introductory distributed database concepts and its structures.
Describe terms related to distributed object database design and management.
Produce the transaction management and query processing techniques in DDBMS.
Relate the importance and application of emerging database technology.

Text Books: Stefano Ceri and Giuseppe Pelagatti, “Distributed Databases Principles and Systems”, McGraw-Hill, 2013

Reference Books:

1. M Tamar Ozsu and Patrick Valduriez, “Principles of Distributed Database Systems”, Pearson Education, 2011
2. Chanda Ray, “Distributed Database Systems”, Pearson Education, 2009

CS 23.464 **Business Intelligence (BI) and its Application** **Total Lectures: 40**

Credit: 3-0-0-3

Prerequisite(s): Basic knowledge of RDBMS (relational database management system) concepts with hands-on exposure (includes design & implementation of table structures).

Objective(s):

- To understand the BI terminologies and framework, basics of data integration (Extraction Transformation Loading).
- To understand the multi-dimensional data modeling, basics of enterprise reporting and application of the concepts using open source/Microsoft tools.

1. **Unit-1:** Introduction to Business Intelligence, Introduction to OLTP and OLAP, BI Definitions & Concepts, Business Applications of BI, BI Framework, Role of Data Warehousing in BI, BI Infrastructure Components – BI Process, BI Technology, BI Roles & Responsibilities. [4]
2. **Unit-2:** Basics of Data Integration (Extraction Transformation Loading), Concepts of data integration need and advantages of using data integration, introduction to common data integration approaches, introduction to ETL using SSIS, Introduction to data quality, data profiling concepts and applications [12]
3. **Unit-3**: Introduction to Multi-Dimensional Data Modeling, Introduction to data and dimension modeling, multidimensional data model, ER Modeling vs. multi-dimensional modeling, concepts of dimensions, facts, cubes, attribute, hierarchies, star and snowflake schema, introduction to business metrics and KPIs, creating cubes using SSAS [6]
4. **Unit-4:** Basics of Enterprise Reporting**,** Introduction to enterprise reporting, concepts of dashboards, balanced scorecards, introduction to SSRS Architecture, enterprise reporting using SSRS [12]
5. **Unit 5:** Relevant lab exercises to get exposure to BI concepts & tool, and Seminars from the topics related to Business Intelligence space [6]

Outcome(s): Upon successful completion of the course, students will be able to

1. Differentiate between Transaction Processing and Analytical applications and describe the need for Business Intelligence
2. Demonstrate understanding of technology and processes associated with Business Intelligence framework
3. Demonstrate understanding of Data Warehouse implementation methodology and project life cycle
4. Given a business scenario, identify the metrics, indicators and make recommendations to achieve the business goal
5. Design an enterprise dashboard that depicts the key performance indicators which helps in decision making
6. Demonstrate application of concepts in Microsoft BI suite

Software Requirements: SQL Server along with Business Intelligence Development Studio (BIDS)- Version 2008

Books:

1. David Loshin “Business Intelligence” The Savvy Manager’s 2nd Edition, October 2012.
2. Mike Biere, “Business intelligence for the enterprise” 1st Edition, Prentice

Hall Professional, 2003.
3. Larissa Terpeluk Moss, S. Atre, "Business Intelligence Roadmap: The Complete Project Lifecycle for Decision-support Applications" 1st, Addison-Wesley Professional, 2003.
4. Cindi Howson, "Successful Business Intelligence: Secrets to Making BI Applications", 1st Edition, McGraw-Hill Education, 2008.
5. Brian Larson, "Delivering Business Intelligence with Microsoft SQL Server 2008", 2nd Edition, McGraw-Hill Education, 2008.
6. Lynn Langit, "Foundations of SQL Server 2005 Business Intelligence" Apress, 1st Edition, 2007
7. Stephen Few, "Information Dashboard Design: Effective Visual Communication of Data", 1st edition, O'Reilly, 2006

CS 23.431 **Advanced Web Technology** **Total Lectures: 40**
Credit:3-1-0-4

Prerequisite(s): None

Objective(s):

- To learn the platform neutral fundamentals of secure, dynamic web application development.
- To learn how to implement a web application using one specific set of open source server side tools: PHP and MySQL.

1. **JavaScript and XML:** Client side scripting with JavaScript, variables, functions, conditions, loops and repetition, pop up boxes, advance JavaScript: JavaScript and objects, JavaScript own objects, the DOM and web, manipulation using DOM, forms and validations, Introduction to XML, XML key components, DTD and Schemas, using XML with application, transforming XML using XSL and XSLT. [7]
2. **Advanced PHP and MySQL**: Server-side web scripting, Adding PHP to HTML, Syntax and Variables, Passing information between pages, Strings, Arrays and Array Functions, Numbers, string, form processing, files, PHP/MySQL Functions, Displaying queries in tables, Building Forms from queries, String and Regular Expressions, Sessions, Type and Type Conversions. [9]
3. **Introduction to Web Services:** Use of Web Services, Types of Web Services, Introduction to Content Management System CMS (Types, Usages, Benefits). [5]
4. **Ajax & jQuery**: Introduction and Use of Ajax in Website. jQuery: Introduction, jQuery UI: Date picker, auto complete, tooltip, accordion, retrieving page content, manipulating page content, working with events. [8]
5. **Introduction to Bootstrap & Node.js:** Components of Bootstrap, Node.js: Node Package Manager (NPM), Node.js, Webserver – Server and Clients. **React**: Introduction to ReactJS, Environment Setup, JSX, Components, State, Props, Validating Props, Component API, Component Life Cycle, Forms, Events. [13]

Outcome(s):

Upon completion of the course students will be able to:
Use the JavaScript to develop the dynamic web pages.
Use server side scripting with PHP to generate the web pages dynamically using the database connectivity.
Develop the modern Web applications using the client and server side technologies and the web design fundamentals.

Text Books:

1. Luke Welling, Laura Thomson: PHP and MySQL Web Development, Pearson, 2016.
2. Steven Holzner, “PHP: The Complete Reference”, TATA McGraw Hill, 2015.
3. Alex Young, “Node.js in Action”, 2ed, Bradley Meck
4. Robin Nixon: Learning PHP, MySQL, and JavaScript, O'Reilly Media, 2012.

Reference Books:

1. Elizabeth Naramore, Jason Gerner, Yann Le Scouarnec, Jeremy Stolz, MichaelK. Glass, Beginning PHP5, Apache, and MySQL Web Development, Wrox, 2005.
2. Developing Web Applications in PHP and AJAX, Harwani, McGrawHill, 2010.

3. W. Jason Gilmore: Beginning PHP and MySQL 5 From Novice to Professional, Apress, 2010.
4. Harwani, "Developing Web Applications in PHP and AJAX", McGrawHill
5. HTML 5, Black Book, Dreamtech Press