SCHOOL
OF
INFORMATION AND COMMUNICATION TECHNOLOGY

COURSE STURCTURE

5 YEARS DUAL DEGREE B.TECH. (CSE) + M. TECH. / MBA

SPECIALIZATION:

SOFTWARE ENGINEERING

GAUTAMBUDDHAUNIVERSITY
GAUTAM BUDH NAGAR, GREATER NOIDA
2014-2015

5 Year Dual Degree B. Tech. (Computer Science & Engineering)+ M.Tech. / MBA

Effective from: 2014 -2015 **Software Engineering**

SEMESTER I

Sr. No.	Courses Code	Courses	L-T-P	Credits
1	CY101 / PH102	Engineering Chemistry / Engineering Physics	3-1-0	4
2	MA101	Engineering Mathematics – I	3-1-0	4
3	CE101	Engineering Mechanics	2-1-0	3
4	CS101	Computer Programming – I	2-0-0	2
5	EC101 / EE102	Basic Electronics / Electrical Technology	2-0-0	2
6	HU101	English Proficiency	2-0-0	2
7	SS101	Human Values & Buddhist Ethics	2-0-0	2
8	CY103 / PH104	Engg. Chemistry Lab / Engg. Physics Lab	0-0-2	1
9	CS 181	Computer Programming Lab1	0-0-3	2
10	CS183	Cyber Security Lab	0-0-3	2
11	EC181 / EE104	Basic Electronics Lab / Electrical Technology Lab	0-0-2	1
12	GP101	General Proficiency	------	1
		Total Credits		**26**
		Total Contact Hours	**16-3-19 =**	**29**

SEMESTER II

Sr. No.	Courses Code	Courses	L-T-P	Credits
1	PH102 / CY101	Engineering Physics/ Engineering Chemistry	3-1-0	4
2	MA102	Engineering Mathematics – II	3-1-0	4
3	CE106	Ecology and Environment	2-1-0	3
4	CS102	Computer Programming – II	2-0-0	2
5	EE102 / EC101	Electrical Technology / Basic Electronics	2-0-0	2
6	HU102	Professional Communication	2-0-0	2
7	SS102	History of Science & Technology	2-0-0	2
8	PH104 / CY103	Engg. Physics Lab/ Engg. Chemistry Lab	0-0-2	1
9	CS182	Computer Programming Lab - II	0-0-3	2
10	EE104 / EC181	Electrical Technology Lab/ Basic Electronics Lab	0-0-2	1
11	ME102	Engineering Workshop	0-0-3	2
12	GP102	General Proficiency	------	1
		Total Credits		**26**
		Total Contact Hours	**16-3-10 =**	**29**

5 Year Dual Degree B. Tech. (Computer Science & Engineering)+ M.Tech. / MBA

Effective from: 2014 -2015 **Software Engineering**

SEMESTER – III

Sr. No.	Courses Code	Courses	L-T-P	Credits
1	MA201	Engineering Mathematics III	3-1-0	4
2	EC201	Digital Electronics	3-0-0	3
3	CS201	Internet Technology	3-0-0	3
4	CS203	Operating Systems	3-0-0	3
5	CS205	Data Structure & Algorithms	2-1-0	3
6	CS207	System Analysis and Design	3-0-0	3
7	EC281	Digital Electronics Lab	0-0-3	2
8	CS281	Internet Technology Lab	0-0-3	2
9	CS283	Data Structure and Algorithms Lab	0-0-3	2
10	GP201	General Proficiency	------	1
		Total Credits		**26**
		Total Contact Hours	**17-2-9 =**	**28**

SEMESTER – IV

Sr. No.	Courses Code	Courses	L-T-P	Credits
1	MA202	Quantitative Techniques	3-1-0	4
2	EC210	Principles of Communication	2-1-0	3
3	CS202	Software Engineering	3-0-0	3
4	CS204	Discrete Structure	3-1-0	4
5	CS206	Data Base Management System	3-0-0	3
6	CS208	Principles of Programming Language	2-0-0	2
7	EC282	Analog Communication Lab	0-0-3	2
8	CS282	Software Engineering Lab	0-0-3	2
9	CS284	Database Management System Lab	0-0-3	2
10	GP202	General Proficiency	------	1
		Total Credits		**26**
		Total Contact Hours	**16-3-9 =**	**28**

5 Year Dual Degree B. Tech. (Computer Science & Engineering)+ M.Tech. / MBA

Effective from: 2014 -2015 **Software Engineering**

SEMESTER – V

Sr. No.	Courses Code	Courses	L-T-P	Credits
1	CS301	Theory of Automata	3-1-0	4
2	CS303	Web Development	3-1-0	4
3	CS305	Computer Graphics	3-1-0	4
4	CS307	Analysis and Design of Algorithms	3-1-0	4
5	CS309	Computer Organization & Architecture	3-0-0	3
6	CS 311	E Commerce	2-0-0	2
7	CS381	Web Development Lab	0-0-3	2
8	CS383	Computer Graphics Lab	0-0-3	2
9	GP301	General Proficiency	------	1
		Total Credits		**26**
		Total Contact Hours	**17-4-6 = 27**	

SEMESTER – VI

Sr. No.	Courses Code	Courses	L-T-P	Credits
1	CS302	Distributed Operating System	3-1-0	4
2	CS304	Concepts of Artificial Intelligence	3-1-0	4
3	CS306	Advanced Computer Architecture	3-1-0	4
4	CS308	Computer Networks	3-1-0	4
5	EC304	Microprocessor and Interfacing	3-0-0	3
6	CS 312	Management Information System	2-0-0	2
7	EC384	Microprocessor and Interfacing Lab	0-0-3	2
8	CS382	Computer Networks Lab	0-0-3	2
9	GP302	General Proficiency	------	1
		Total Credits		**26**
		Total Contact Hours	**17-4-6 = 27**	

5-Year Dual Degree B. Tech. (Computer Science & Engineering)+ M.Tech. / MBA

Effective from: 2014 -2015 **Software Engineering**

SEMESTER – VII

Sr. No.	Courses Code	Courses	L-T-P	Credits
1	SS401	Social Aspects of Engineering	2-1-0	3
2	CS401	Compiler Design	3-1-0	4
3	CS403	Object-Oriented Analysis	3-0-0	3
4	CS405	Formal Methods	3-1-0	4
5	CS 527	Research Technique in ICT	3-0-0	3
6	CS481	Compiler Design Lab	0-0-3	2
7	CS491	Seminar	0-0-2	2
8	CS493	Minor Project	0-0-8	4
9	GP401	General Proficiency	------	1
		Total Credits		**26**
		Total Contact Hours	**14-3-13 =**	**30**

SEMESTER VIII

Sr. No.	Courses Code	Courses	L-T-P	Credits
1	MA402	Modeling & Simulation	3-1-0	4
2	CS532	Advanced Software Engineering	3-0-0	3
3	CS534	Open Source Software Systems	3-0-0	3
4	CS536	Object-Oriented Software Engineering	3-0-0	3
5		**Elective-1**	3-0-0	3
6	CS586	Object-Oriented Software Engineering Lab	0-0-3	2
7	CS592	Major Project	0-0-10	5
8	GP532	General Proficiency	------	1
		Total Credits		**24**
		Total Contact Hours	**15-1-13 =**	**29**

Electives (1)		
1	CS542	Software Architecture and Design
3	CS546	Software Re-Engineering
4	CS548	Software Reusability
5	CS550	Web-Based Software Engineering

5-Year Dual Degree B. Tech. (Computer Science & Engineering)+ M.Tech. / MBA

Effective from: 2014 -2015 **Software Engineering**

Summer Semester (After 8th Semester)				
Sr. No.	**Courses Code**	**Courses**	**L-T-P**	**Credits**
1	CS490	Project	0-0-20	10
		Total Credits	**10**	
		Total Contact Hours	0-0-10 =	**20**

SEMESTER IX

Sr. No.	Courses Code	Courses	L-T-P	Credits
1	CS631	Software Testing	3-0-0	3
2	CS633	Component-Based Software Engineering	3-0-0	3
3		**Electives –2**	3-0-0	3
4		**Electives –3**	3-0-0	3
5	CS681	Software Testing Lab	0-0-3	2
6	CS691	Dissertation Part - I	0-0-14	7
7	GP631	General Proficiency	------	1
		Total Credits		**22**
		Total Contact Hours	**12-0-17 =**	**29**

Electives (2&3)		
1	CS641	Aspect-Oriented Software Engineering
2	CS643	Software Reliability
3	CS645	Software Quality Assurance
4	CS647	Software Maintenance
5	CS649	Software Performance
6	CS 651	Software Measurement and Estimation

SEMESTER X

Sr. No.	Courses Code	Courses	L-T-P	Credits
1	CS690	Dissertation Part – II	------	21
2	GP632	General Proficiency	------	1
		Total Credits		**22**

GRAND TOTAL CREDITS = 255

(SEMESTER - I)

COMPUTER PROGRAMMING – I			
Course Code:	**CS101**	**Credits:**	**2**
No. of Lectures (Hrs/Week):	**2**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**30**	**End Sem Exam Hours:**	**3**

UNIT I INTRODUCTION TO COMPUTER

Definition, characteristic, generation of computers, basic components of a computer system, memory, input, output and storage units, hard copy devices, high level language and low level language, software, system software, application software, hardware, firmware, compiler, interpreter and assembler.

UNIT II INTRODUCTION TO PROGRAMMING CONCEPT

Introduction to algorithm and flow chart; representation of algorithm using flow chart symbol, pseudo code, basic algorithm design, characteristics of good algorithm, development of algorithm.

UNIT III INTRODUCTION TO C PROGRAMMING LANGUAGE

declaring variables, preprocessor statements, arithmetic operators, programming style, keyboard input , relational operators, introduction, feature of C language, concepts, uses, basic program structure, simple data types, variables, constants, operators, comments, control flow statement :if, while, for, do-while, switch .

UNIT IV

User defined data types, arrays, declaration and operations on arrays, structure, member accessing, structure and union, array of structures, functions, declaration and use of functions, parameter passing, dynamic memory allocation.

UNIT V FUNDAMENTALS OF POINTERS

Declaration and usages of pointers, operations that can be performed on computers, use of pointers in programming exercises, parameter passing in pointers, call by value, call by references, Introduction to LINUX: LINUX structure, directory, LINUX commands.

Text Books:

1. C Programming, Herbert Shield
2. Programming in ANSI C by E. Balagurusamy, Tata Mgraw Hill,

Reference Books:

3. C Programming Language 2nd Edition by Brian, W Kernighan Pearson Education.
4. C. Puzzle Book: Puzzles For The C. Programming Language by Alan R Feuer Prentice Hall- Gale
5.Expert C Programming: Deep C Secrets (s) by Peter Van Der Linden Dorling Kindersley India.
6. Introduction To UNIX System by Morgan Rachel Tata Mcgraw Hill Education.
7. C: A Reference Manual (5th Edition) by Samuel P. Harbison&Samuel P. Harbison.
8. Programming Using the C Language by Hutchison,R.C, Mcgraw Hill Book Company, New York

COMPUTER PROGRAMMING LAB – I			
Course Code:	**CS181**	**Credits:**	**2**
No. of Lectures (Hrs/Week):	**3**		
Total No. of Lectures:	**10**	**End Sem Exam Hours:**	**3**

EXPERIMENTS:

1. Write a C program to reverse a given number, find the sum of digits of the number.
2. Write a C program to concatenate two strings.
3. Write a C program to take marks of a student as input and print the his/her grade bases on following criteria using if – else statements

Marks <40 FAIL

40<= Marks <59 GOOD

59 <= Marks < 80 Excellent

80 <= Marks Outstanding

4. Perform experiment 3 using switch case statement.
5. Write a C program to compute the length of a string using while loop.
6. Write a C program to convert all the lowercase letter to uppercase letter and all uppercase letters to lower case letter given a string as input.
7. Write a C program to compute the roots of a quadratic equation.
8. Write a C program to check whether a given number is prime or not, also check whether it is divisible by a number k or not.
9. Write a C program to check whether a given year is leap year or not.
10. Write a C program to take two matrixes as input and print the sum of two matrixes.
11. Write a C program to display the address of a variable using pointer.
12. Write a C program to compute the length of a string using pointer.
13. Create a structure called STUDENT having name, registration number, class, session as its field. Compute the size of structure STUDENT.
14. Write a C program to check weather a given string is palindrome or not.
15. Write a C program to generate following patterns.

```
        1
      2   2
    3   3   3
  4   4   4   4
```

```
        A
      B   B
    C   C   C
  D   D   D   D
```

BASIC ELECTRONICS LAB			
Course Code:	**EC181**	**Credits:**	**2**
No. of Lab (Hrs/Week):	**1**	**End Sem Exam Hours:**	**3**
Total No. of Lab Sessions:	**15**		

List of Experiments

1. Study of Multimeter and Function Generator /Counter.
2. Study of Cathode-Ray Oscilloscope.
3. To calculate the Equivalent Resistance of the Series and parallel resistive network.
4. To calculate the Equivalent Capacitance of the Series and parallel capacitive network.
5. To Plot the V-I Characteristics of P-N Junction Diode in forward bias and reverse bias.
6. To study the working of a P-N Junction Diode as a switch.
7. To plot the V-I Characteristics of a Zener Diode.
8. To plot the input and output waveforms of clipper circuits.
9. Study the Half wave rectifier.
10. Study of Full wave Bridge Rectifier.
11. Study of Centre Tapped Full Wave Rectifier.
12. To plot the input and output characteristic of transistor's Common Base configuration.
13. To plot the input and output characteristic of transistor's Common Emitter configuration.
14. To plot the input and output characteristic of transistor's Common Collector configuration.
15. To verify the truth table of various logic gates.

(SEMESTER - II)

COMPUTER PROGRAMMING – II			
Course Code:	**CS102**	**Credits:**	**2**
No. of Lectures (Hrs/Week):	**2**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**30**	**End Sem Exam Hours:**	**3**

UNIT I OBJECT-ORIENTED PROGRAMMING

Concept of object-oriented programming (OOP), benefits of OOP, application of OOP, Java history, Java features, Java streaming, Java and Internet, Java contribution to Internet: Java applets, security, portability; Java environment, Java library, Java program structure, Java program, Java Virtual Machine (JVM) architecture, Just In Time compiler (JIT), data type, variables and arrays, operators, control statements, object-oriented paradigms; abstraction, encapsulation, inheritance, polymorphism, Java class and OOP implementation

UNIT II DATA TYPE, OPERATORS AND CONTROL STATEMENT

Data types, Java key words, identifiers, constants, variables, declaration and scope of the variable, symbolic constant, type casting, arithmetic operator, relational operator, logical operator, assignment operator, increment and decrement operator, conditional operator, bitwise operator, ?: operator, arithmetic expressions, expressions, type conversions in expressions, mathematical functions, more data types: arrays, strings, vectors, wrappers classes, program control statements: decision making and branching: if, if….else, else….if, else if ladder, switch, decision making and looping: while, do….while, for.

UNIT III CLASSES, OBJECTS AND METHODS

Java class libraries, class fundamentals, object, methods, adding variables, add methods, creating objects, accessing class members, constructors, methods overloading, static members, nesting of methods, inheritance: extending a class, overriding methods, final variables and methods, final classes, finalizer methods, abstract methods and classes, visibility control, exception handling fundamental.

UNIT IV INTERFACES AND PACKAGES

Interfaces, extending interfaces, implementing interfaces, interfaces references, accessing interface variable, creating queue interface, variable in interfaces, packages, finding a packages and classpath, package and member access, Java API package, system package, naming conventions, creating package, accessing a package, adding a class to a package, hiding classes,

UNIT V MULTITHREADING AND APPLET PROGRAMMING

Multithreading programming: creating threads, thread class and runnable interface extending the thread class, stopping and blocking a thread, life cycle of a thread, thread methods, thread exceptions, thread priority, synchronization, thread communication using notify(), wait(), and notify all(), applet programming : applet basic, applets architecture, a complete applet skeleton, building applets code, applets life cycle, creating a executable applet, designing a web page, applets tag, passing parameters to applets, applets and HTML.

Reference Books:

1. Programming with JAVA, E. Balagurusawamy, Tata McGraw Hill, 1998.
2. JAVA Beginner's guide, Herbert Schildt, Tata McGraw Hill, 2007.
3. Java How to Program, Deitel & Deitel, Prentice-Hall, 1999.
4. The Complete Reference JAVA 2, Herbert Schildt, 5th Edition, Tata McGraw Hill, 2002.
5. The Complete Reference JAVA 2, Herbert Schildt, 7th Edition, Tata McGraw Hill, 2009.
6. The Java Programming Language, Ken Arnold, James Gosling, Addison-Wesley, 1996.
7. How to Program Java, Peter Coffee, Ziff-Davis Press, 1996.

Basic Electronics			
Course Code:	**EC 101**	**Credits:**	**2**
No. of Lectures (Hrs/Week):	**2**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**30**	**End Sem Exam Hours:**	**3**

Unit I: Passive Components Resistances, Capacitors and Inductors, Component Specifications, Applications, Response to dc and sinusoidal voltage/current excitations Semiconductor Theory Metals, Insulators and Semiconductor materials, energy band diagram, Intrinsic and Extrinsic Semiconductors, Doping, Fermi level, Fermi level of P-type and N-type materials, Mobility, Drift Current and Diffusion Current. Current conduction in Semiconductors, Generation and Recombination of Charges .

Unit II: Semiconductor DiodesTheory of P-N Junction, Ideal & Practical diode, Concept of AC and DC Resistances, V-I Characteristics, Diode Equivalent Circuits, Transition and Diffusion Capacitance, Reverse Recovery Time, Zener and Avalanche breakdown, Tunnel Diodes, Varactor Diode, Light Emitting Diode

Unit III: Diode Applications and Wave Shaping Circuits Load line analysis, series and parallel combinations, Half wave & Full wave Rectifiers, Clippers & Clampers.

Unit IV: Transistors Bipolar Junction Transistor- Construction, Operation, Transistor Configurations, Input and Output Characteristics, AC and DC Load line, operating point, Effect of shifting the operating point. Biasing, Thermal Runaway, Effect of temperature on the characteristics, Early effect, introduction to JFET and MOSFET

Unit V: Logic Gates and Operational Amplifiers
Binary number, Digital systems, Boolean algebra, logic gates, logic functions, realization of logic gates by
electronic devices, Positive and negative logic, representation of binary numbers, half adder, full adder, flip-flops, Op-Amp, Practical Op-Amp, Open loop and closed loop configurations, Applications of OpAmps as inverting and non-inverting amplifier

Text Books:
[1] Boyelasted an Nashlsky: Electronics Devices and circuit Theory, TMH.
[2] Gayakwad :Op-Amps and Linear Inegrated Circuits , PHI.

References:
[1] Millman & Halkias :Integrated Electronics ,TMH.
[2] Morris Mano: Digital Design ,PHI.
[3] Malvino :Electronics Principles,TMH.

Basic Electronics LAB			
Course Code:	**EC 181**	**Credits:**	**2**
No. of Lectures (Hrs/Week):	**2**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**30**	**End Sem Exam Hours:**	**3**

List of Experiments

1. Study of Multimeter and Function Generator /Counter.
2. Study of Cathode-Ray Oscilloscope.
3. To calculate the Equivalent Resistance of the Series and parallel resistive network.
4. To calculate the Equivalent Capacitance of the Series and parallel capacitive network.
5. To Plot the V-I Characteristics of P-N Junction Diode in forward bias and reverse bias.
6. To study the working of a P-N Junction Diode as a switch.
7. To plot the V-I Characteristics of a Zener Diode.
8. To plot the input and output waveforms of clipper circuits.
9. Study the Half wave rectifier.
10. Study of Full wave Bridge Rectifier.
11. Study of Centre Tapped Full Wave Rectifier.
12. To plot the input and output characteristic of transistor‟s Common Base configuration.
13. To plot the input and output characteristic of transistor‟s Common Emitter configuration.
14. To plot the input and output characteristic of transistor‟s Common Collectorconfiguration.
15. To verify the truth table of various logic gates

COMPUTER PROGRAMMING-II LAB			
Course Code:	**CS182**	**Credits:**	**2**
No. of Practical (Hrs/Week):	**3**		
Total No. of Lab Sessions:	**10**	**End Sem. Exam Hours:**	**2**

Programs/Experiments List:

1. Write a separate Java Code to implement each of the following:
   Class, Command Line Argument, how to enter value through keyboard

2. Write a separate Java Code to implement each of the following data types:
   Variable, Constant, Arrays, Strings, Vectors, Wrappers Classes, Type Casting

3. Write a separate Java Code to implement each of the following operators:
   Arithmetic operator, Relational operator, Logical operator, Assignment operator, Increment & Decrement operator, Conditional operator, Bitwise operator, ?: operator

4. Write a separate Java Code to implement each of the following control statements:
   Decision statement, Loops statement and Branch statements

5. Write a separate Java Code to implement each of the following sorting:
   Bubble Sort, Selection Sort, Insertion Sort, Merge Sort

6. Write a separate Java Code to implement each of the following:
   Class, Object, Constructors, Method, Method Overloading and Method Overriding

7. Write a separate Java Code to implement each of the following:
   Final variable, final class, final method, abstract class, abstract method and concrete method

8. Write a separate Java Code to implement each of the following OOPs concepts:
   Abstraction, Polymorphism, Encapsulation, Inheritance
9. Write a separate Java Code to implement each of the following:
   Exception handling with Try, Catch, Throw, Throws, Finally
   Multiple catch statement with the following exceptions :
   ArithmeticException, ArrayOutOfBoundsException and ArrayStoreException

10. Write a separate Java Code to implement each of the following:
    Visibility Controls: Private, Public and Protected

11. Write a separate Java Code to implement each of the following:
    Interface, extending and implementing interface

12. Write a separate Java Code to implement each of the following:
    Multithreading: Create thread with thread class and runnable interface, thread priorities, synchronization

13. Write a separate Java Code to implement each of the following:
    Packages : Create package A with following methods and import this package A into another Java program to show the result of methods of package A.
    i) First method: Factorial number with the help of recursion.
    ii) Second method: Fibonacci Series

iii) Third Method: Generate first 10 prime numbers and show the sum of first 10 prime numbers.

14. Write Java Code to generate the following output on applet with the help of two dimensional array and show the result with the help of HTML file.

7 14 21 28 35 42 49 56 63 70	Sum = 385
5 10 15 20 25 30 35 40 45 50	Sum = 275
3 6 9 12 15 18 21 24 27 30	Sum = 165

15. Write a Java Code to design the following web page with the help of applet and HTML.

School of Information and Communication Technology
GAUTAM BUDDHA UNIVERSITY
GREATER NOIDA

- **Student Name:**
- **Enrollment Number:**
- **Programme Name:**
- **Semester:**
- **Course Name:**
- **E-Mail ID:**
- **Mobile Number:**
- **Blood Group:**

(SEMESTER - III)

QUANTITATIVE TECHNIQUES			
Course Code:	**MA201**	**Credits:**	**4**
No. of Lectures (Hrs/Week):	**3+1**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45+15**	**End Sem Exam Hours:**	**3**

UNIT-I

Random Events, Independent and dependent events, axioms of the theory of probability, Simple and conditional probability, Bayes theorem.

UNIT-II

One dimensional random variables (discrete and continuous), distribution of a random variable (density function and cdf), Characteristic function of a random variable and its utility, Bivariate random variable, joint, marginal and conditional distributions, joint characteristic function, Moments, Moment Generating functions, Skewness, Kurtosis.

UNIT-III

Bernoulli, Binomial, Poisson, Geometric, Uniform, Exponential, Normal, Earlang, Weibull, Method of least squares (Fitting of straight lines, Polynomials, Exponential and logarithmic curves), covariance and correlation.

UNIT-IV

Sampling theory (small and large), Test of hypothesis and significance: Chi-square test, t-test, z-test, F-test, Questionaire design.

UNIT-V

Markov chain, Chapman- Kolmogorov Equation, Classification of states.

Text Books:

1. T. Veerarajan. Probability, Statistics and Random Processes, Tata McGraw-Hill.

Reference Books:

2. V. K. Rohatgi: An Introduction to Probability Theory and Mathematical Statistics. John Wiley & Sons 1976.
3. John Freund: Introduction to Probability. Dover Publications.
4. Marylees Miller, John E. Freund, Irwin Miller: John E. Freund's Mathematical Statistics: With Applications, Prentice Hall, 2003.
5. Levin and Rubin, Statistics for Management, Prentice Hall.

DIGITAL ELECTRONICS			
Course Code:	**EC201/EC431**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45**	**End Sem Exam Hours:**	**3**

UNIT I

Number systems & codes, Binary arithmetic Boolean algebra and switching function. Minimization of switching function, concept of prime implecant etc. Karnaugh's map method, Quine & McCluskey's method, cases with don't care terms and multiple outputs switching function. Logic gates, NAND, NOR realization of switching function; half-adder half-subtractor full-adders full-subtractor circuits. Series & parallel addition and BCD adders, look-ahead carry generator.

UNIT II

Linear wave shaping circuits, Bistable, monostable & astable multivibrators, Schmitt trigger circuits .Introduction to D/A converters. Various types of Analog to Digital & Digital to Analog converters sample & hold circuits and V-F converters.

UNIT III

Logic families: RTL, DTL, all types of TTL circuits, ECL, 12 L and PMOS, NMOS &CMOS logic etc. Gated flip-flops and gated multivibrators etc; Interfacing between TTL to MOS, vice-versa.

UNIT IV

Introduction to shift registers / ring counters synchronous & asynchronous counters and designing of combinational circuits like code converters & counters etc.

UNIT V

Semiconductor memories & designing with ROM and PLA: Decoders Encoders multiplexers & demultiplexers**.**

Text Books:

1. Tocci, "Digital Systems Principles & Applications".
2. M. Mano, "Digital Logic & Computer Design", (PHI).

Reference Books:

3. John F. Wakerly, Digital Design: Principles & Practices, Pearson Education.2003
4. Richard F.Tinder, Engineering Digital Design, 2/e, Harcourt India Private Ltd., 2001
5. William I. Fletcher, An Engineering Approach to Digital Design, Pearson Education
6. William H.Gothmann, Digital Electronics: An Introduction to Theory and Practice, Eastern Economy Edition, Prentice-Hall of India Private Limited, New Delhi. 2001.
7. Jacob Millman & Herbert Taub,Pulse,Digitaland Switching Waveforms,13th Reprint,Tata McGraw Hill Publishing Company Ltd.,NewDelhi, 1999

INTERNET TECHNOLOGY			
Course Code:	**CS201**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45**	**End Sem Exam Hours:**	**3**

UNIT I OVERVIEW OF INTERNET ANDWEB

Introduction to internet, history of Internet and web, Internet services and accessibility, uses of internet, Internet standards, Internet protocols- IP, TCP,UDP and host names, web server, proxy server, fast ready connections on the web, web browsers, Netscape communication suite, Microsoft Internet explorer, firewalls, data security.

UNIT II WEB DESIGN

Key issues in web site design, introduction to HTML, SGML- DTD, DTD elements, attributes, outline of an HTML document, head section- prologue, link, base, meta, script, style, body section- headers, paragraphs, text formatting, linking, internal linking, embedding images, lists, tables, frames, other special tags and characters, XHTML, XML, structuring data, XML schema documents, document object model, security and management issues for creating a website.

UNIT III BROWSING SYSTEMS

Searching and web casting technique, popular web servers, basic features, bookmarks, cookies, progress indicators, customization of browsers, browsing tricks, next generation web browsing, search engines, architecture of search engines, search tools, web crawlers, types of crawlers, scalable web crawler, incremental crawler, parallel crawler, focused crawler, agent based crawler, case study of IE, counters, Internet chat, hardware and software requirements for Internet and web based applications, Internet and web technologies.

UNIT IV JAVASCRIPT

Introduction, Language elements, objects of JavaScript, other objects like data, math, string, regular expressions, arrays.

UNIT V ACTIVE SERVER PAGES

Creating interactive applications using active server pages : client and server side script in C#, variables and constants, creating modules, creating objects from classes, ASP's object model, arrays, collections, control structures, using request and response objects, integration with database.

Reference Books:

1. Raj Kamal, Internet and Web Technologies, TMH, 2005.
2. Monica D'Souza, Web publishing, TMH, 2001.
3. David Crowder and Rhonda Crowder, Web Design, IDG Books India, 2001.
4. Musciano C., HTML and XHTML the Definitive Guide, 6th edition, OReilly, 2006.
5. Deitel H., Deitel P., Internet and World Wide Web: How to Program, 4 edition, PHI.

OPERATING SYSTEMS			
Course Code:	**CS203/CS431**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45**	**End Sem Exam Hours:**	**3**

UNIT I INTRODUCTION TO OPERATING SYSTEM

Importance of operating systems, basic concepts and terminology about operating system, memory management functions, processor management functions, device management functions, information management functions.

UNIT II PROCESS MANAGEMENT

Elementary concept of process, job scheduler, process scheduling, operation on process, threads, overview, scheduling criteria, scheduling algorithms, algorithm evaluation process synchronization, synchronization hardware, semaphores, classical problem of synchronization, monitors and atomic transaction deadlocks: system model, deadlock characterization, deadlocks prevention, deadlocks avoidance, deadlocks detection, recovery from deadlock.

UNIT III MEMORY MANAGEMENT

Memory management, logical versus physical address space, swapping, contiguous allocation, paging, segmentation, demand paging, page replacement, page replacement algorithms, allocation of frames, thrashing, demand segmentation.

UNIT IV STORAGE MANAGEMENT

File concept, directory structure, protection, file-system structure, allocation method, free-space management, directory implementation.

UNIT V I/O SYSTEMS

I/O hardware, Application of I/O interface, Overview of Kernel I/O subsystem, three types of I/O systems, memory based I/O, I/O based I/O, peripheral based I/O.

Reference Books:

1. Galvin, Wiley, Operating Systems Concepts, 8th edition, 2009.
2. James L Peterson, Operating Systems Concept, John Wiley & Sons Inc, the 6Rev edition, 2007.
3. Deitel H. M., An Introduction to Operating Systems, Addison-Wesley, 1990.
4. Stallings William, Operating Systems, PHI, New Delhi, 1997.
5. Madnick and Donavon, Operating Systems, McGraw Hill, International edition, 1978.
6. S. Tanenbaum Modern Operating Systems, , Pearson Education, 3rd edition, 2007.
7. Nutt, Operating System, Pearson Education, 2009.
8. S. Tanenbaum, Distributed Operating Systems, Prentice Hall, 2nd edition, 2007.
9. M. Singhal & N. Shivaratri, Advanced Concepts in Operating Systems, McGraw Hill, 2003.

DATA STRUCTURES AND ALGORITHMS			
Course Code:	**CS205**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45**	**End Sem Exam Hours:**	**3**

UNIT I INTRODUCTION TO DATA STRUCTURES

Abstract data types, sequences as value definitions, data types in C, pointers in C, data structures and C, arrays in C, array as ADT, one dimensional array, Implementing one dimensional array, array as parameters, two dimensional array, structures in C, implementing structures, Unions in C, implementation of unions, structure parameters, allocation of storage and scope of variables, recursive definition and processes: factorial function, fibonacci sequence, recursion in C, efficiency of recursion, hashing: hash function, open hashing, closed hashing: linear probing, quadratic probing, double hashing, rehashing, extendible hashing.

UNIT II STACK, QUEUE AND LINKED LIST

Stack definition and examples, primitive operations, example -representing stacks in C, push and pop operation implementation, queue as ADT, C Implementation of queues, insert operation, priority queue, array implementation of priority queue, inserting and removing nodes from a list-linked implementation of stack, queue and priority queue, other list structures, circular lists: stack and queue as circular list - primitive operations on circular lists, header nodes, doubly linked lists, addition of long positive integers on circular and doubly linked list.

UNIT III TREES

Binary trees: operations on binary trees, applications of binary trees, binary tree representation, node representation of binary trees, implicit array representation of binary tree, binary tree traversal in C, threaded binary tree, representing list as binary tree, finding the Kth element, deleting an element, trees and their applications: C representation of trees, tree traversals, evaluating an expression tree, constructing a tree.

UNIT IV SORTING AND SEARCHING

General background of sorting: efficiency considerations, notations, efficiency of sorting, exchange sorts: bubble sort; quick sort; selection sort; binary tree sort; heap sort, heap as a priority queue, sorting using a heap, heap sort procedure, insertion sorts: simple insertion, shell sort, address calculation sort, merge sort, radix sort, sequential search: indexed sequential search, binary search, interpolation search.

UNIT V GRAPHS

Application of graph, C representation of graphs, transitive closure, Warshall's algorithm, shortest path algorithm, linked representation of graphs, Dijkstra's algorithm, graph traversal, traversal methods for graphs, spanning forests, undirected graph and their traversals, depth first traversal, application of depth first traversal, efficiency of depth first traversal, breadth first traversal, minimum spanning tree, Kruskal's algorithm, round robin algorithm.

Text Books:

1. Aaron M. Tenenbaum, Yeedidyah Langsam, Moshe J. Augenstein, 'Data structures using C', Pearson Education, 2004 / PHI.

References Books:

2. E. Balagurusamy, 'Programming in Ansi C', Second Edition, TMH, 2003.
3. Robert L. Kruse, Bruce P. Leung Clovis L.Tondo, 'Data Structures and Program Design in C', Pearson Education, 2000 / PHI.

SYSTEM ANALYSIS AND DESIGN			
Course Code:	**CS207**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45**	**End Sem Exam Hours:**	**3**

UNIT I DATAANDINFORMATION

Types of information: operational, tactical, strategic and statutory, why do we need information systems, management structure, requirements of information at different levels of management, functional allocation of management, requirements of information for various functions, qualities of information, small case study.

UNIT II SYSTEMS ANALYSIS AND DESIGN LIFE CYCLE

Requirements determination, requirements specifications, feasibility analysis, final specifications, hardware and software study, system design, system implementation, system evaluation, system modification, role of systems analyst, attributes of a systems analyst, tools used in system analysis

UNITIII INFORMATION GATHERING

Strategies, methods, case study, documenting study, system requirements specification, from narratives of requirements to classification of requirements as strategic, tactical, operational and statutory.

UNIT IV FEASIBILITY ANALYSIS

Deciding project goals, examining alternative solutions, cost benefit analysis, quantifications of costs and benefits, payback period, system proposal preparation for managements, parts and documentation of a proposal, tools for prototype creation.

UNIT V TOOLS FOR SYSTEMS ANALYSTS

Data flow diagrams, case study for use of DFD, good conventions, leveling of DFDs, leveling rules, logical and physical DFDs, software tools to create DFDs, decision tables for complex logical specifications, specification oriented design vs procedure oriented design

Reference Books:

1. System Analysis and Design – Elias M.Awad.
2. System Analysis and Design –Perry Edwards
3. Analysis and Design of Information Systems – James A.Senn

DIGITAL ELECTRONICS LAB			
Course Code:	**EC281/EC483**	**Credits:**	**2**
No. of Practical (Hrs/Week):	**3**		
Total No. of Lab Sessions:	**10**	**End Sem. Exam Hours:**	**2**

Programs/Experiments List:

1. Verify the truth table of AND Gate.
2. Verify the truth table of OR and NOT gates.
3. Verify the truth table of NAND , NOR Gates.
4. Verify the truth table of and EX-OR Gate.
5. Design a combinational circuit to realize the function f(ABC)=A(B+C) using NAND gates.
6. Design a half adder using NOR gates only
7. Design full adder
8. Design a given size of Mux using gates.
9. Verify RS and JK flip flops
10. Verify D and T Flip flops
11. Design a up/down 3-bit counter
12. Design a 3 bit shift register.

INTERNET TECHNOLOGY LAB			
Course Code:	**CS281**	**Credits:**	**2**
No. of Lectures (Hrs/Week):	**3**		
Total No. of Lectures:	**10**	**End Sem Exam Hours:**	**2**

List of Experiments

1. Design a web page using XHTML which shows the Hyperlinks, List, Tables.
2. Design a web page in which users insert the username and password using XHTML form.
3. Design a web page using XHTML to insert an image and adding a marquee.
4. Write XML schema document for Book domain.
5. Design a website using HTML, XHTML and JavaScript.
6. Using Java Script display a line of text in a web page.
7. Using Java Script display a text in an alert Dialog box.
8. Using Java Script , take input of two integer from the user and display the result in the web page.
9. Using Java Script, Show the equality and relational operators.
10. Write a script that reads integers and determines the square root of the integer using square root method of Math object.
11. Write a script which uses the control statements.
12. Write a script that reads five integers and determines the largest and the smallest integers in the group.
13. Write a code to show connectivity between SQL server and ASP.net.
14. Introduction to different types of tools using in ASP.net Technology.
15. Design a web page using ASP.net.

DATA STRUCTURE AND ALGORITHMS LAB			
Course Code:	**CS283**	**Credits:**	**2**
No. of Lectures (Hrs/Week):	**3**		
Total No. of Lectures:	**10**	**End Sem Exam Hours:**	**2**

List of Experiments

1. Run time analysis of Fibonacci Series
2. Study and Application of various data Structure
3. Study and Implementation of Array Based Program
    a. Searching (Linear Search, Binary Search)
    b. Sorting (Bubble, Insertion, Selection, Quick, Merge etc)
    c. Merging
4. Implementation of Link List
    a. Creation of Singly link list, Doubly Linked list
    b. Concatenation of Link list
    c. Insertion and Deletion of node in link list
    d. Splitting the link list into two link list
5. Implementation of STACK and QUEUE with the help of
    a. Array
    b. Link List
6. Implementation of Binary Tree, Binary Search Tree, Height Balance Tree
7. Write a program to simulate various traversing Technique
8. Representation and Implementation of Graph
    a. Depth First Search
    b. Breadth First Search
    c. Prims Algorthim
    d. Kruskal's Algorithms
9. Implementation of Hash Table

(SEMESTER - IV)

NUMERICAL METHODS FOR ANALYSIS			
Course Code:	**MA202**	**Credits:**	**4**
No. of Lectures (Hrs/Week):	**3+1**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45+15**	**End Sem Exam Hours:**	**3**

UNIT-I

Zeroes of transcendental and polynomials, Bisection method, Regula-falsi method and Newton-Raphson method, Secant Method, Rate of convergence of above methods.

UNIT-II

Interpolation, Finite differences, difference tables, Newton's forward and backward interpolation, Divided differences, Lagrange's and Newton's divided difference formula for unequal intervals, Error analysis,Least square approximation.

UNIT-III

Numerical differentiation, Numerical integration, Error Analysis, Newton-Cote quardrature formulae, Trapezoidal, Simpson's one third and three-eight rules.

UNIT IV

Solution of system of linear equations, (Direct and Indirect Methods) Gauss- Seidal method, Gauss-Jordan Method, Crout method, Gauss-elimination and LU-Decomposition, successive-overrelaxation, Power method for largest eigenvalue, Jacobi method for real symmetric matrices.

UNIT V

Numerical Solutions of ODE and PDE: Runge-Kutta and predictor corrector methods for IVPs. Finite difference methods for BVPs, Numerical solutions of parabolic and elliptic partial differential equations, Finite element method.

Text Books:

1. Gerald C.F., Wheatley P.O., Applied Numerical Analysis, 6th Ed., Pearson Education, 1999.
2. Gupta R.S., Elements of Numerical Analysis, 1st Ed., Macmillan 2009.

Reference Books:

1. Jain M.K., Iyen S.R.K., Jain R.K.: Numerical Methods for Scientific and Engineering Computation 5th Ed., New Age International, New Delhi, 2007.
2. Smith G.D. : Numerical Solution of Partial Differential Equations, 2nd Ed., Oxford, 1978.

PRINCIPLES OF COMMUNICATION			
Course Code:	**EC210**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**2+1**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**30+15**	**End Sem Exam Hours:**	**3**

UNIT I LINEAR MODULATION

Modulation techniques: need and types of modulation techniques, amplitude modulation, frequency spectrum, power distribution, generation and detection of AM, comparision of various AM systems, synchronous detection technique, error in synchronous detection, SSB signal detection, Frequency Division Multiplexing(FDM).

UNIT I IANGLE MODULATION

Frequency and phase modulation, frequency spectrum, bandwidth requirement, frequency and phase deviation, modulation index, narrowband FM, wide band FM & their spectrum, FM modulators: Direct (Parameter Variation Method) and Indirect (Armstrong) method of frequency modulation. FM Detector: slope detector, foster seely discriminator, ratio detector and PLL detectors, pre-emphasis & de-emphasis, capture effect.

UNIT III SAMPLING AND RECONSTRUCTION:

Sampling of signal, sampling theorem for low pass and band pass signals, aliasing, Pulse Amplitude Modulation (PAM), Time Division Multiplexing (TDM), types of sampling, PAM, PWM and PPM, Pulse Code Modulation: quantization, quantization error, signal-to-noise ratio in PCM, companding, line coding, Differential PCM (DPCM), Delta Modulation (DM), Adaptive Delta Modulation (ADM), slope overload error, granular noise.

UNIT IV DIGITAL MODULATION TECHNIQUES :Amplitude Shift Keying (ASK), Binary Phase Shift Keying (BPSK), Differential Phase Shift Keying (DPSK), Quadrature Phase Shift Keying (QPSK), M-ary PSK, Binary Frequency Shift Keying (BFSK), M-ary FSK, Minimum Shift Keying, Quadrature Amplitude Modulation (QAM), probability of error, Bit Error Rate (BER), comparison of digital modulation techniques on the basis of probability of error, matched filter.

UNIT V INFORMATION THEORY AND CODING:

Information,amount of information, unit of information, average information or entropy, information rate, joint and conditional entropy, discrete memory less channel, special channels, mutual information and channel capacity, mutual information and channel capacity for special channels, shannon's theorem, shannon- hartley theorem, bandwidth & s/n trade off, source coding, code length and code efficiency, source coding theorem, fixed length codes, variable length codes, source coding techniques: Shannon Fano and Huffman coding algorithms, linear block codes, systematic linear blocks codes, parity check matrix, syndrome testing, cyclic code, hamming code, error detection and correction codes, convolution codes

Text Books:

1. Lathi B.P., Analog and Digital Communication systems, 3/E Oxford Press, 2007
2. Taub & Schilling, Principles of communication systems, 3/E McGraw Hill, 2000.
3. B. Sklar, Digital Communication, Pearson Education

Reference Books:

4. Taub & Schilling, Principles of Communication system, TMH.
5. Singh and Sapre: Communication System, TMH
6. Proakis and Salehi, Fundamentals of Communication Systems, 1/E Pearson Education, 2005.
7. Hwei P Hsu,Analog and Digital Communication Schaum Series TMH, 2 Edition.
8. Tomasi: Advanced Electronics Communication Systems, 6th Edition, PHI
9. Couch: Digital and Analog Communication, Pearson Education

SOFTWARE ENGINEERING			
Course Code:	**CS202/CS433**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45**	**End Sem Exam Hours:**	**3**

10. Carlson, Communication Systems, 5/E McGraw Hill, 2004.

UNIT I SOFTWARE ENGINEERING

Introduction to software engineering: definitions, role of software engineering, planning a software project, defining the problem, developing a solution strategy, planning the development process, software engineering process paradigms, principles of software engineering, software engineering activities.

UNIT II REQUIREMENT ANALYSIS AND DESIGN

Software Requirement Specification (SRS): Introduction, need of SRS, significance, characteristics of SRS, Structure of SRS, IEEE standards for SRS design, functional and non-functional requirements, Requirement gathering and analysis, requirement engineering and management.

UNIT III SOFTWARE DESIGN PROCESS

Software Design: Introduction, design process activities: architectural design, Abstract specification, Interface design, component design, data structure design, algorithm design modular approach, top-down design, bottom-up design, design methods: data-flow model: data flow diagram, entity-relation-attribute model: E-R diagram, structural model: structure charts, context diagrams, objectmodels: use case modeling, use case diagrams, sequence diagrams, cohesion and coupling.

UNIT IV SOFTWARE LIFE CYCLE MODELS

Software Development Life Cycle (SDLC), SDLC models, waterfall model and its variations, prototype model, iterative enhancement model, spiral model, RAD model, comparison of these models, software development teams, software development environments, validation and traceability, maintenance, prototyping requirements, Software project management.

UNIT V SOFTWARE TESTING AND MAINTENANCE

Testing Methods: unit testing, integration testing, system testing, acceptance testing, testing techniques: white box testing, black box testing, thread testing, regression testing, alpha testing, beta testing, static testing, dynamic testing, Evolution of software products, economics of maintenance, category of software maintenance, Role of product development life cycle, deployment model, adaptive maintenance, corrective maintenance, perfective maintenance, enhancement request, proactive defect prevention, problem reporting, problem resolution, software maintenance from customers' perspective, maintenance standard: IEEE-1219, ISO-12207.

Reference Books:

1. Pankaj Jalote, An Integrated Approach to Software Engineering, Narosa Publishing House, New Delhi 1997.
2. Ian Sommerville, Software Engineering, Pearson Education, 2009.
3. Pressman Roger S., Software Engineering: Practitioner's Approach, McGraw-Hill Inc., 2004.
4. Software Engineering: Software Reliability, Testing and Quality Assurance, Nasib S. Gill, Khanna Book Publishing Co (P) Ltd., New Delhi, 2002.

DISCRETE STRUCTURE			
Course Code:	**CS204**	**Credits:**	**4**
No. of Lectures (Hrs/Week):	**3+1**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45+15**	**End Sem Exam Hours:**	**3**

UNIT I MATHEMATICAL LOGIC

Statements and notations, connectives, well formed formulas, truth tables, tautology, equivalence implication, normal forms, predicates: predicative logic, free & bound variables, rules of inference, consistency, proof of contradiction, automatic theorem proving.

UNIT II SET THEORY

Properties of binary relations, equivalence, compatibility and partial ordering relations, hasse diagram. functions: inverse function comports of functions, recursive functions, lattice and its properties, pigeon hole principles and its application, algebraic structures: algebraic systems examples and general properties, semi groups and monads, groups sub groups' homomorphism, isomorphism.

UNIT III ELEMENTARY COMBINATORICS

Basis of counting, combinations & permutations, with repetitions, constrained repetitions, binomial coefficients, binomial multinomial theorems, the principles of inclusion – exclusion.

UNIT IV RECURRENCE RELATION

Generating functions, function of sequences calculating coefficient of generating function, recurrence relations, solving recurrence relation by substitution and generating funds, characteristics roots solution of in homogeneous recurrence relation.

UNIT V GRAPH THEORY

Representation of graph, DFS, BFS, spanning trees, planar graphs. graph theory and applications, basic concepts isomorphism and sub graphs, multi graphs and euler circuits, hamiltonian graphs, chromatic numbers

Text Books :

1. Discrete and Combinational Mathematics- An Applied Introduction-5th Edition – Ralph. P.Grimaldi, Pearson Education
2. Discrete Mathematical Structures with applications to computer science Trembly J.P. & Manohar .P, TMH
3. Discrete Mathematics and its Applications, Kenneth H. Rosen, Fifth Edition.TMH.

Reference Books:

4. Discrete Mathematics with Applications, Thomas Koshy, Elsevier
5. Discrete Mathematical Structures, Bernand Kolman, Roberty C. Busby, Sharn Cutter Ross, Pearson Education/PHI.
6. Discrete Mathematical structures Theory and application-Malik & Sen
7. Discrete Mathematics for Computer science, Garry Haggard and others, Thomson.
8. Discrete Mathematics for Computer Scientists & Mathematicians, J.L. Mott, A. Kandel, T.P. Baker Prentice Hall.
9. Logic and Discrete Mathematics, Grass Man & Trembley, Person Education.

DATABASE MANAGEMENT SYSTEM			
Course Code:	**CS206/CS436**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45**	**End Sem Exam Hours:**	**3**

UNIT I DATA BASE SYSTEM

Data base system vs. file system, view of data, data abstraction, instances and schemas, data models, ER model, relational model, database languages, DDL, DML, database access for applications programs, data base users and administrator, transaction management, data base system structure, storage manager, query processor, history of data base systems, data base design and ER diagrams, beyond ER design entities, attributes and entity sets, relationships and relationship sets, additional features of ER model, concept design with the ER model, and conceptual design for large enterprises.

UNIT II RELATIONAL MODEL

Introduction to the relational model, integrity constraint over relations, enforcing integrity constraints, querying relational data, and logical data base design, destroying /altering tables and views. relational algebra and calculus: relational algebra, selection and projection set operations, renaming, joins, division, relational calculus, tuple relational calculus, domain relational calculus, expressive power of algebra and calculus.

UNIT III BASIC SQL QUERY

Examples of basic SQL queries, nested queries, correlated nested queries set, comparison operators, aggregative operators, NULL values, comparison using null values, logical connectivity's, AND, OR and NOTR, impact on SQL constructs, outer joins, disallowing NULL values, complex integrity constraints in SQL triggers and active data bases.

UNIT IV SCHEMA REFINEMENT

Problems caused by redundancy, decompositions, problem related to decomposition, reasoning about FDS, FIRST, SECOND, THIRD normal form, BCNF, forth normal form, lossless join decomposition, dependency preserving decomposition, schema refinement in data base design, multi valued dependencies.

UNIT V OVERVIEW OF TRANSACTION MANAGEMENT

ACID properties, transactions and schedules, concurrent execution of transaction, lock based concurrency control, performance locking, and transaction support in SQL, crash recovery, concurrency control, Serializability and recoverability, lock management, lock conversions, dealing with dead locks, specialized locking techniques, concurrency without locking, crash recovery: ARIES, log, other recovery related structures, the write, ahead log protocol, check pointing, recovering from a system crash, media recovery, other approaches and interaction with concurrency control.

References Books:

1. Elmasri Navrate, Data Base Management System, Pearson Education, 2008.
2. Raghurama Krishnan, Johannes Gehrke, Data Base Management Systems, TMH, 3rd edition, 2008.
3. C. J. Date, Introduction to Database Systems, Pearson Education, 2009.
4. Silberschatz, Korth, Database System Concepts, McGraw hill, 5^{th} edition, 2005.
5. Rob, Coronel & Thomson, Database Systems Design: Implementation and Management, 2009.

PRINCIPLES OF PROGRAMMING LANGUAGE			
Course Code:	**CS208**	**Credits:**	**2**
No. of Lectures (Hrs/Week):	**2**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**30**	**End Sem Exam Hours:**	**3**

UNIT I INTRODUCTION

Characteristics of programming Languages, Factors influencing the evolution of programming language, developments in programming methodologies, desirable features and design issues. Programming language processors: Structure and operations of translators, software simulated computer, syntax, semantics, structure, virtual computers, binding and binding time.

UNIT II ELEMENTARY AND STRUCTURED DATA TYPES

Data object variables, constants, data types, elementary data types, declaration, assignment and initialization, enumeration, characters, strings. Structured data type and objects: Specification of data structured types, vectors and arrays, records, variable size data structure, pointers and programmer constructed data structure, Sets files. Sub Program and programmer defined data types: Evolution of data types, abstractions, encapsulations, information hiding, sub programms, abstract data types.

UNIT III SEQUENCE CONTROL

Implicit and Explicit sequence control, sequence control with within expression and statements, recursive sub programmes, exception handling, co routines, scheduled sub programmes, concurrent execution. Data control referencing environments, static and dynamic scope, local data local data referencing environment, shared data: explicit common environment dynamic scope parameter passing mechanism.

UNIT IV STORAGE MANAGEMENT

Major run time requirements, storage management phases, static storage management, stack based, heap based storage management. Syntax and translation: General syntactic criteria, syntactic element of a language, stages in translation, formal syntax and semantics.

UNIT V OPERATING ANDPROGRAMMING ENVIRONMENT

Batch processing environments, embedded system requirements, theoretical models, introduction to functional programming, lambda calculus, data flow language and object oriented language, comparison in various general and special purpose programming languages e.g. fortran, C, Pascal, Lisp, etc.

Reference Books:

1. Terrance W Pratt, "Programming Languages: Design and Implementation" PHI
2. Sebesta, "Concept of Programming Language", Addison Wesley
3. E Horowitz, "Programming Languages", 2nd Edition, Addison Wesley
4. "Fundamentals of Programming Languages", Galgotia

SOFTWARE ENGINEERING LAB			
Course Code:	**CS282**	**Credits:**	**2**
No. of Lab (Hrs/Week):	**3**		
Total No. of Lab Sessions:	**10**	**End Sem Exam Hours:**	**2**

List of Experiments

1. Introduction and project definition.
2. Software process overview with configuration management tool.
3. Design the software requirements by using Requisite Pro.
4. Introduction to UML and use case diagrams with the help of Rational Rose.
5. System modelling and design of DFD and ER diagram.
6. Design of Flow of events and activity diagram by using Rational Rose.
7. OO analysis and discovering classes with the help of Requisite Pro.
8. Design the Interaction diagrams, sequence and collaboration diagrams with the help of software engineering tool.
9. Software architecture and object-oriented design by using Rational Rose.
10. Draw the traceability matrix with the help of designing the requirements and feature matrix

DATABASE MANAGEMENT SYSTEM LAB			
Course Code:	**CS284**	**Credits:**	**3**
No. of Lab (Hrs/Week):	**3**	**End Sem Exam Hours:**	**3**
Total No. of Lab Sessions:	**10**		

List of Experiments

1. Introduction to MySQL, an exercise of data types in MySQL & Data Definition Language Commands
2. Exercise on Data Manipulation Language and Transaction Control Commands
3. Exercise on Types of Data Constraints
4. Exercise on JOINS (Single-Table) Using Normalization
5. Exercise on JOINS (Multiple-Table) Using Normalization
6. Exercise on GROUP BY/ORDER BY Clause and Date Arithmetic
7. Exercise on different Functions (Aggregate, Math and String)
8. Exercise on different types of sub queries
9. Procedures
10. View
11. Triggers

(SEMESTER - V)

Theory of Automata			
Course Code:	**CS301**	**Credits:**	**4**
No. of Lectures (Hrs/Week):	**3+1**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45+15**	**End Sem Exam Hours:**	**3**

UNIT I AUTOMATA

Introduction; alphabets, strings and languages; automata and grammars, deterministic finite automata (DFA)-formal definition, simplified notation: state transition graph, transition table, language of DFA, Nondeterministic finite Automata (NFA), NFA with epsilon transition, language of NFA, equivalence of NFA and DFA, minimization of finite automata, distinguishing one string from other, Myhill-Nerode Theorem

UNIT II REGULAR EXPRESSIONS AND LANGUAGES

Regular expression (RE), definition, operators of regular expression and their precedence, algebraic laws for regular expressions, Kleen‟s theorem, regular expression to FA, DFA to regular expression, arden theorem, non regular languages, pumping lemma for regular languages. application of pumping lemma, closure properties of regular languages, decision properties of regular languages, FA with output: moore and mealy machine, equivalence of moore and mealy machine, applications and limitation of FA.

UNIT III CONTEXT-FREE GRAMMAR AND LANGUAGES

Context Free Grammar (CFG) and Context Free Languages (CFL): definition, examples, derivation, derivation trees, ambiguity in grammar, inherent ambiguity, ambiguous to unambiguous CFG, useless symbols, simplification of CFGs, normal forms for CFGs: CNF and GNF, closure properties of CFLs, decision properties of CFLs: emptiness, finiteness and membership, pumping lemma for CFLs.

UNIT IV PUSH DOWN AUTOMATA

Push Down Automata (PDA): description and definition, instantaneous description, language of PDA, acceptance by final state, acceptance by empty stack, deterministic PDA, equivalence of PDA and CFG, CFG to PDA and PDA to CFG, two stack PDA

UNIT V TURING MACHINES (TM)

Basic model, definition and representation, instantaneous description, language acceptance by TM, variants of turing machine, TM as computer of integer functions, universal TM, church‟s thesis recursive and recursively enumerable languages, halting problem, introduction to undecidability, undecidable problems about TMs. Post Correspondence Problem (PCP), modified PCP, introduction to recursive function theory.

References Books:

1. Hopcroft, Ullman, “Introduction to Automata Theory, Languages and Computation”, Pearson Education
2. K.L.P. Mishra and N.Chandrasekaran, “Theory of Computer Science : Automata, Languages and Computation”, PHI
3. Martin J. C., “Introduction to Languages and Theory of Computations”, TMH
4. Papadimitrou, C. and Lewis, C.L., “Elements of the Theory of Computation”, PHI

WEB DEVELOPMENT			
Course Code:	**CS303**	**Credits:**	**4**
No. of Lectures (Hrs/Week):	**3+1**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45+15**	**End Sem Exam Hours:**	**3**

UNIT I INTRODUCTION TO .NET

.NET, advantages of .NET, Common Language Runtime (CLR), CLR architecture, Just-in-time complier, Microsoft Intermediate Language, IL with IL Disassembler, framework, types and version of framework, common class library, common type system, common language specifications, languages under .NET.

UNIT II LANGUAGE FUNDAMENTALS

Basic programming rules, data types, variable declaration and initialization, using the *using* and *imports* keywords, literals, unicode characters and strings, operators, conditional statements, looping statements, arrays, structures, concept of class and objects, creating and using class library, creating and using namespaces, oops paradigm: encapsulation, abstraction, polymorphism, inheritance.

UNIT III VISUAL BASIC PROGRAMMING

Creating interactive applications using active server pages: client and server side script in C#, creating modules, creating objects from classes, flow control and exception handling, working with windows and web forms, mouse event, hiding and displaying controls, button control, label control, TextBox control, radio button control, check box control, list box control, using requestand response objects, integration with database, ADO.NET.

UNIT IV JAVA SERVER PAGES (JSP)

Evolution of web application, overview of Hyper Text Transfer Protocol, servlet, servlet life cycle, servlet classes, threading models, HTTP session, Java Server Pages, JSP syntax and sementics, expression, scriptlets and declaration, request dispatching, session and thread management.

UNIT V JSP APPLICATIONS AND JDBC

Develop and deploy web application with JSP, JSP and XML, JSP testing and debugging, JDBC,

Text Books:

1. Hanna, The Complete Reference JSP 2.0, Tata McGraw Hill, 2003.
2. Mike Mcgrath, Java Server Pages, Dreamtech Press, 2009.
3. Visual Basic 2008 Programming, Black Book, Dreamtech Press, 2009.

Reference Books:

8. E. Balagurusawamy, Programming with JAVA, Tata McGraw Hill, 1998.
9. Christian Nagel, Professional C# and .NET 4, Wrox, 2010.
10. Karli Watson, Beginning Microsoft Visual C#, Wrox, 2010.

COMPUTER GRAPHICS			
Course Code:	**CS305**	**Credits:**	**4**
No. of Lectures (Hrs/Week):	**3+1**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45+15**	**End Sem Exam Hours:**	**3**

UNIT I INTRODUCTION

Application areas of Computer Graphics, overview of graphics systems, video-display devices, raster-scan systems, random scan systems, graphics monitors and work stations and input devices, Output primitives : Points and lines, line drawing algorithms, mid-point circle and ellipse algorithms. Filled area primitives: Scan line polygon fill algorithm, boundary-fill and flood-fill algorithms

UNIT II 2-D GEOMETRICAL TRANSFORMS

Translation, scaling, rotation, reflection and shear transformations, matrix representations and homogeneous coordinates, composite transforms, transformations between coordinate systems, 2-D viewing : The viewing pipeline, viewing coordinate reference frame, window to view-port coordinate transformation, viewing functions, Cohen-Sutherland and Cyrus-beck line clipping algorithms, Sutherland –Hodgeman polygon clipping algorithm

UNIT III REPRESENTATION AND TRANSFORMATION

3-D object representation Polygon surfaces, quadric surfaces, spline representation, Hermite curve, Bezier curve and B-Spline curves, Bezier and B-Spline surfaces, basic illumination models, polygon rendering methods, 3-D Geometric transformations: Translation, rotation, scaling, reflection and shear transformations, composite transformations, 3-D viewing: Viewing pipeline, viewing coordinates, view volume and general projection transforms and clipping

UNIT IV VISIBLE SURFACE DETECTION METHODS

Classification, back-face detection, depth-buffer, scan-line, depth sorting, BSP-tree methods, area sub-division and octree methods.

UNIT V COMPUTER ANIMATION

Design of animation sequence, general computer animation functions, raster animation, computer animation languages, key frame systems, motion specifications.

References Books :

1. “Computer Graphics C version”, Donald Hearn and M.Pauline Baker, Pearson Education.
2. “Computer Graphics Principles & practice”, second edition in C, Foley, VanDam, Feiner and Hughes, Pearson Education.

ANALYSIS AND DESIGN OF ALGORITHMS			
Course Code:	**CS307**	**Credits:**	**4**
No. of Lectures (Hrs/Week):	**3+1**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45+15**	**End Sem Exam Hours:**	**3**

UNIT I BASIC CONCEPTS OF ALGORITHMS

Introduction, notion of algorithm, fundamentals of algorithmic solving, important problem types, fundamentals of the analysis framework, asymptotic notations and basic efficiency classes.

UNIT II MATHEMATICAL ASPECTS AND ANALYSIS OF ALGORITHMS

Mathematical analysis of non-recursive algorithm, mathematical analysis of recursive algorithm, example: fibonacci numbers, empirical analysis of algorithms, algorithm visualization.

Unit III ANALYSIS OF SORTING ANDSEARCHING ALGORITHMS

Brute force, selection sort and bubble sort, sequential search and brute-force string matching, divide and conquer, merge sort, quick sort, binary search, binary tree, traversal and related properties, decrease and conquer, insertion sort, depth first search and breadth first search.

UNIT IV ALGORITHMIC TECHNIQUES

Transform and conquer ,presorting, balanced search trees, avl trees, heaps and heap sort, dynamic programming, Warshall's and Floyd's algorithm, optimal binary search trees, greedy techniques, Prim's algorithm, Kruskal's algorithm, Dijkstra's algorithm, Huffman trees.

UNIT V ALGORITHM DESIGN METHODS

Backtracking, n-Queen's problem, Hamiltonian circuit problem, subset-sum problem, branch and bound, assignment problem, knapsack problem, traveling salesman problem.

Text Books:

1. Anany Levitin, "Introduction to the Design and Analysis of Algorithm", Pearson Education Asia, 2003.

References Books:

2. T.H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein, "Introduction to Algorithms", PHI Pvt. Ltd., 2001
3. Sara Baase and Allen Van Gelder, "Computer Algorithms - Introduction to Design and Analysis", Pearson Education Asia, 2003.
4. A.V.Aho, J.E. Hopcroft and J.D.Ullman, "The Design and Analysis of Computer Algorithms", Pearson Education Asia, 2003.

COMPUTER ORGANIZATION & ARCHITECTURE			
Course Code:	**CS309/CS438**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45**	**End Sem Exam Hours:**	**3**

UNIT I COMPUTER ARITHMETIC AND NUMBER SYSTEM

Number representation; number system, fixed and floating point number representation, arithmetic algorithms (addition, subtraction, booth multiplication).

UNIT II REGISTER TRANSFER AND MICROOPERATION

Register transfer language, bus and memory transfers, bus architecture, bus arbitration, arithmetic logic, shift microoperation, arithmetic logic shift unit, design of fast address.

UNIT II PROCESSOR DESIGN

Processor organization: general register organization, stack organization, addressing mode, instruction format, data transfer & manipulations, program control, reduced instruction set computer.

UNIT IV INPUT-OUTPUT ORGANIZATION

I/O interface, synchronous and asynchronous data transfer, strobe, handshaking schemes, modes of transfer, interrupts & interrupt handling, direct memory access, input-output processor.

UNIT V MEMORY ORGANIZATION

Memory hierarchy, main memory (RAM and ROM Chips), organization of 2d and $2^{1/2}$ d, auxiliary memory, cache memory, virtual memory, memory management hardware.

Reference Books:

1. Patterson, Computer Organisation and Design, Elsevier Pub. 2009
2. William Stalling, " Computer Organization", PHI
3. Vravice,Hamacher & Zaky, "Computer Organization", TMH
4. Mano," Computer System Architecture", PHI
5. John P Hays, " Computer Organization", McGraw Hill
6. Tannenbaum," Structured Computer Organization', PHI
7. P Pal chaudhry, ' Computer Organization & Design', PHI

(SEMESTER - VI)

DISTRIBUTED OPERATING SYSTEM			
Course Code:	**CS302**	**Credits:**	**4**
No. of Lectures (Hrs/Week):	**3+1**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45+15**	**End Sem Exam Hours:**	**3**

UNIT I INTRODUCTION TO DISTRIBUTED SYSTEMS

Distributed systems : goals hardware concepts software, design, communication distributed systems: layered protocol: ATM networks client server model, remote procedure call, group communication.

UNIT II SYNCHRONIZATION

Clock synchronization, mutual exclusion, election atomic transactions, dead locks, process and processors: threads, system models processor, Allocation, scheduling fault tolerance, real time distributed systems.

UNIT III DISTRIBUTED FILE SYSTEMS

File system design and implementation, trends in distributed file systems.

UNIT IV SHARED MEMORY

Introduction, bus based multi processors ring based multiprocessors switched multiprocessors - NUMA comparison of shared memory systems.

UNIT V

consistency models, page based distributed shared memory, shared variable distributed shared memory, object based distributed shared memory.

Reference Books:

1. Andrew S.Tanenbaum: Distributed Operating System, Prentice Hall International Inc. 1995.
2. A.S. Tanenbaum, Modern Operating Systems, Pearson Education Asia, 2001.
3. M. Singhal and N. G. Shivaratri, Advance Concepts in Operating Systems, McGraw-Hill, 1994.
4. J. W. S. Liu, Real-Time Systems, Pearson Education, 2000.

CONCEPTS OF ARTIFICIAL INTELLIGENCE			
Course Code:	**CS304**	**Credits:**	**4**
No. of Lectures (Hrs/Week):	**3+1**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45+15**	**End Sem Exam Hours:**	**3**

UNIT I INTRODUCTION TO ARTIFICIAL INTELLIGENCE

Basic concept of artificial intelligence (AI), history of AI, AI and consciousness, weak and strong AI, physical symbol system hypothesis, comparison of computer and human skills, practical systems based on AI, development of logic, components of AI.

UNIT II PROBLEM SOLVING THROUGH AI

Defining problem as state space search, analyzing the problem, representing the problems from AI viewpoint, production system, developing production rules, characteristics of production system, algorithm for problem solving using AI technique.

UNIT III SEARCH TECHNIQUES

Use of search in AI problem solution, blind search techniques, heuristic search techniques, concept of heuristic knowledge, designing of the heuristic function, types of heuristic search techniques: generate and test, best first search, problem reduction using AND – OR graph, local search technique, branch and bound search, memory bounded search technique, local beam search, properties of heuristic search techniques, overestimation and underestimation of heuristic function, hill climbing search, simulated annealing search, constraint satisfaction, means ends analysis.

UNIT IV INTRODUCTION TO LOGIC

Introduction, propositional calculus, syntax of propositional calculus, semantics of propositional calculus, well formed formula, properties of statements, inferencing of propositional logic, predicate logic, syntax of predicate logic, semantics of predicate logic, representation of facts First Order Predicate Logic (FOPL), inferencing in predicate logic, concept of resolution, resolution algorithm, skolemization, Types of resolution, unit resolution, binary resolution.

UNIT V PROLOG and LISP

Basic concept of programming languages related to artificial intelligence problems, concept of programming in Logic, basic prolog constructs, atoms, defining the rules, writing small programs in prolog, concept of list processing, basic LISP constructs, writing functions in LISP, some simple programs of LISP.

Reference books:

1. Artificial Intelligence, Elanie Reich: Tata mcgraw Hill publishing house, 2008.
2. Artificial intelligence, Peterson, TataMcGraw Hill, 2008.
3. Artificial intelligence, Russel and Norvig, Pearson Printice Hall Publication, 2006.
4. Artificial Intelligence, Winston, PHI publication, 2006.

ADVANCED COMPUTER ARCHITECTURE			
Course Code:	**CS306**	**Credits:**	**4**
No. of Lectures (Hrs/Week):	**3+1**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45+15**	**End Sem Exam Hours:**	**3**

UNIT I INTRODUCTION TO PARALLEL PROCESSING

Parallelism in uniprocessor system, basic uniprocessor architecture, parallel processing mechanism, balancing of sub system bandwidth, multiprogramming and time sharing, parallel computer structures, pipeline computers, array computers, multiprocessor systems, dataflow computer concept, architectural classification scheme: multiplicity of instruction-data streams, serial versus parallel processing, parallelism versus pipelining, parallel processing applications, productive modeling simulation, engineering design and automation.

UNIT II PRINCIPLES OF PIPELINING AND VECTOR PROCESSING

Pipelining- an overlapped parallelism, principles of linear pipelining, clock period, efficiency, throughput, classification of pipeline processors, general pipeline and reservation tables.

UNIT III PRINCIPLES OF DESIGNING PIPELINE PROCESSORS

Effect of branching , data buffering and bussing structures, internal forwarding and register tagging, hazard detection and resolution, job sequencing and collision prevention, reservation and latency analysis, collision free scheduling, state diagram, greedy cycle, pipeline schedule optimization, pipeline throughput, pipeline efficiency.

UNIT IV STRUCTURE ANDALGORITHM FOR ARRAY PROCESSORS

SIMD array processor, SIMD computer organization, inter –PE communication, SIMD interconnection network, static versus dynamic networks, cube interconnection network, shuffle-exchange omega networks, parallel algorithms and SIMD matrix multiplication.

UNIT V MULTIPROCESSOR ARCHITECTURE AND SCHEDULING

Functional structure, loosely coupled and tightly coupled multiprocessor, deterministic scheduling strategy, deterministic scheduling model, control flow versus data flow computer, data flow graphs and languages.

Reference Books

1. Kai Hwang, “Advanced Computer Architecture”, Tata McGrawHill Edition
2. Kai Hwang and Faye A. Briggs, “Computer Architecture and Parallel Processing”, McGraw-Hill International Edition
3. Richard Y. Kain, “Advanced computer architecture: a systems design”, Prentice Hall.
4. James M. Feldman, Charles T. Retter, “Computer architecture: a designer's text based on a generic RISC”, McGraw-Hill
5. Jurij Silc, Borut Robic, Theo Ungerer, “Processor Architecture: From Dataflow to Superscalar and Beyond”, Springer.
6. Hennessy and Patterson, “Computer Architecture: A Quantitative Approach”, Elsevier.
7. Dezso and Sima, “Advanced Computer Architecture”, Pearson.
8. Quinn, “Parallel Computing: Theory & Practice”, TMH.
9. Quinn, “Parallel Programming in C with MPI and Open MP”, TMH

COMPUTER NETWORKS			
Course Code:	**CS308**	**Credits:**	**4**
No. of Lectures (Hrs/Week):	**3+1**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45+15**	**End Sem Exam Hours:**	**3**

UNIT I INTRODUCTION AND PHYSICAL LAYER

Key concepts of computer network, transmission media, network devices, network topology, topology design issues, types of network: LAN, MAN, WAN, PAN, ISDN systems and ATM network, OSI-reference model, open system standards, characteristics of network, TCP/IP model, protocols and standards, encoding technique.

UNIT II SWITCHING ANDDATALINK LAYER

Circuit switching, packet switching, message switching, hybrid switching, and ATM switching, multiplexing techniques: TDMA, FDMA, WDMA, CDMA, data link layer: LLC &MAC level protocols and design issues, issues IEEE 802 LAN Standards, framing, CRC, error control, flow control, HDLC, ALOHA and performance issues. Frames relay networks and performance parameters.

UNIT III NETWORK LAYER

Network layer design issues, overview of IPv4 and IPv6, addressing: class full and classless, static and dynamic, subnet and supernet, autoconfiguration through DHCP, routing protocols: RIP, DVR,LSR, OSFP, BGP, congestion control algorithm, subnet concept, virtual LAN, ICMP, multicasting, mobile IP.

UNIT IV TRANSPORT LAYER

Port addressing schemes, connectionless and connection oriented services: TCP and UDP, wireless TCP, Congestion control, queue management, NAT, PAT, socket format at transport level, socket interface and programming.

UNIT V APPLICATION LAYER

Client server architecture, domain name services, application services: HTTP, TELNET, RLOGIN, FTP, CBR, NFS, SMTP, POP, IMAP, MIME, voice and video over IP.

Text Books:

1. S. Tanenbaum, Computer Networks, 4th edition, Prentice Hall, 2008

References:

2. Data and Computer Communications, W. Stallings, 8th edition, Prentice Hall, 2007
3. TCP/IP Principles, Protocols and Architecture, Douglus E. Comer, Pearson Education
4. F. Haball ,Data Communication, Computer network & open systems - Computer Networks : An Engineering approach - S. Keshav

MANAGEMENT INFORMATION SYSTEM			
Course Code:	**CS 312**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45**	**End Sem Exam Hours:**	**3**

UNIT 1

Foundations Concepts Foundations of Information Systems in Business, The Components of Information Systems, Competing with Information Technology, Fundamentals of Strategic Advantage, Using Information Technology for Strategic Advantage

UNIT 2

Information Technologies Computer Hardware, Computer Systems: End User and Enterprise Computing, Computer Peripherals: Input, Output, and Storage Technologies, Computer Software, Application Software: End-User Applications, System Software: Computer System Management, Data Resource Management, Technical Foundations of Database Management, Managing Data Resources, Telecommunications and Networks, The Networked Enterprise, Telecommunications Network Alternatives

UNIT 3

Business Applications e-Business Systems, Functional Business Systems, Enterprise Business Systems, managing at the Enterprise Level, Enterprise Resource Planning: The Business Backbone, Supply Chain Management: The Business Network, e-Commerce Systems, e-Commerce Fundamentals, e-Commerce Applications and Issues, Supporting Decision Making, Decision Support in Business

UNIT 4

Development Processes Developing Business/IT Strategies, Planning Fundamentals, Implementation Challenges, Developing Business/IT Solutions, Developing Business Systems, Implementing Business Systems,

UNIT 5

Management Challenges Security and Ethical Challenges, Security and Ethical, and Societal Challenges of IT, Security Management of Information Technology,Enterprise and Global Management of Information Technology, Managing Information Technology, Managing Global IT

Text Books:

1. MIS James O'Brien, George M Marakas, Ramesh Behl: Management Information System, Mc Graw Hills .

Reference Book

1. laudon & Laudon : Management Information Systems, Managing the Digital Age, 2007 edition Printice hall
2. O'Brien : Management Information System , 5th Edition, Tata Mc Graw Hills.
3. D. Boody, Boonstra and Kennedy: Managing Information Systems , Pearson

MICROPROCESSORS AND INTERFACING			
Course Code:	**EC304/EC559**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45**	**End Sem Exam Hours:**	**3**

UNIT I

Introduction to 8085 microprocessor: pin diagram, architecture, programming model, instruction set, and classification of instruction set, instruction and data format, timing diagram of instructions, basic concept of programming, addressing modes of 8085 microprocessors.

UNIT II

Microprocessor 8086 architecture, BIU and EU, registers, pin diagram, memory addressing, clock generator 8284, buffers and latches, maximum and minimum modes.

UNIT III

Addressing Modes, Instruction set of 8086, assembly language programming, assemblers, procedures, macros, interrupts, 8086 based multiprocessor systems

UNIT IV

Interfacing Chips- IC 8155 (Static Ram with I/OPorts and Timer), 8755 (EPROM with I/O Ports), 8251A (USART), 8255A (Programmable Peripheral Interface), 8253/8254 (Programmable Interval Timer/Counter), 8257 (DMA Controller), 8259A (Programmable Interrupt Controller).

UNIT V

The 8051 architecture Microprocessor and Microcontroller, Comparison of microprocessors and microcontrollers Microcontroller survey, microcontrollers of different word length, make and features, selection criteria for microcontroller ,8051 microcontroller hardware, I/O pins and internal architecture internal RAM, ROM organization, I/O port circuits ,connecting external memory, addressing modes, instruction set and assembly language programming.

Text Books:

1. A. K. Ray : Advanced Microprocessors and Interfacing, 2nd edition,TMH
2. Mazidi and Mazidi: The 8051 Microcontroller and Embedded Systems, Pearson Education

References:

3. B. B. Brey: The Intel Microprocessors, Architecture, Programming and Interfacing, Pearson Education.
4. Liu Gibson: Microcomputer Systems: The 8086/8088 Family- Architecture, Programming and Design, PHI
5. D. V. Hall: Microprocessors and Interfacing, TMH.
6. Ayala Kenneth:- The 8051 microcontroller, Third Edition, Cengage Learning
7. A. V. Deshmukh: Microcontroller (Theory and Application), TMH.
8. Raj Kamal: Embedded Systems- Architecture, Programming and Design, TMH, New Delhi.
9. R S Gaonkar, Microprocessor, Architecture, Programming, and Applications with the 8085, Penram International Publication, 5/e
10. P.K. Ghosh and P. R. Sridhar, 0000 to 8085 Introduction to microprocessor for Engineers and Scientists, PHI, 2/e

Computer Network Lab	
Course Code: CS 382	**Credits: 2**
No. of Lectures (Hrs/Week): 3hrs/week	**Mid Sem Exam Hours: 2**
Total No. of Lectures:	**End Sem Exam Hours: 3**

1. Introduction to transmission media(CAT5, OFC, COAXIAL CABLE Wireless)
2. Introduces network interfaces(Wired and Wireless)
3. Configure and installing a Ethernet(10/100)
4. Performance evaluation of Ethernet(10/100)
5. Topology design(Ring, Bus)
6. Generation of data packet and measurement(CBR, VBR, Poison)
7. Router configuration
8. Switch configuration
9. Server configuration
10. Congestion control of network
11. QoS of network
12. Protocols and the configuration
13. Wireless systems
14. S3curity (WEP, WPA)
15. Qualnet.

(SEMESTER - VII)

COMPILER DESIGN			
Course Code:	**CS401**	**Credits:**	**4**
No. of Lectures (Hrs/Week):	**3+1**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45+15**	**End Sem Exam Hours:**	**3**

UNIT I INTRODUCTION TO COMPILER

Introduction to compiler, phases and passes, bootstrapping, finite state machines and regular expressions and their applications to lexical analysis, optimization of DFA-based pattern matchers implementation of lexical analyzers, lexical-analyzer generator, LEX-compiler, formal grammars and their application to syntax analysis, BNF notation, ambiguity, YACC, syntactic specification of programming languages: Context free grammars, derivation and parse trees, capabilities of CFG.

UNIT II PARSING TECHNIQUE

Parsers, shift reduce parsing, operator precedence parsing, top down parsing, predictive parsers Automatic construction of efficient parsers: LR parsers, the canonical collection of LR(0) items, constructing SLR parsing tables, constructing canonical LR parsing tables, constructing LALR parsing tables, using ambiguous grammars, an automatic parser generator, implementation of LR parsing tables.

UNIT III SYNTAX-DIRECTED TRANSLATION

Syntax-directed translation schemes, implementation of syntax directed translators, 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 and case statements.

UNIT IV SYMBOL TABLES

Data structure for symbols tables, representing scope information, run-time administration: implementation of simple stack allocation scheme, storage allocation in block structured language, Error detection & recovery: lexical phase errors, syntactic phase errors, semantic errors.

UNIT V CODE GENERATION

Design issues, the target language. addresses in the target code, basic blocks and flow graphs, optimization of basic blocks, code generator. code optimization: machine-independent optimizations, loop optimization, DAG representation of basic blocks, value numbers and algebraic laws, global data-flow analysis

Reference Books:

1. Aho, Sethi & Ullman, "Compilers: Principles, Techniques and Tools", Pearson Education
2. V Raghvan, " Principles of Compiler Design", TMH
3. Kenneth Louden," Compiler Construction", Cengage Learning.
4. Charles Fischer and Ricard LeBlanc," Crafting a Compiler with C", Pearson

OBJECT-ORIENTED ANALYSIS			
Course Code:	**CS403**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45**	**End Sem Exam Hours:**	**3**

UNIT I OBJECT-ORIENTED FUNDAMENTALS

Object model, classes and objects, complexity, classification, notation, process, pragmatics, binary and entity relationship, object types, object state, Object-Oriented Software Development (OOSD) life cycle, foundations of the object-oriented approach, concept of objects, object vs. classes, encapsulation, aggregation, inheritance and polymorphism, de-facto industry standard for general-purpose object-oriented modeling.

UNIT II OBJECT-ORIENTED ANALYSIS

Overview of object analysis: Shater/Mellor, Coad/Yourdon, Rumbaugh, Booch, usecase, conceptual model, behavior, class, analysis patterns, aggregation, advanced principles and strategies for object-oriented analysis, encapsulation and connascence, encumbrance and cohesion, type conformance and closed behavior.

UNIT III OBJECT-ORIENTED METHODS

Unified Modeling Language (UML), diagrams, collaboration, sequence, class, frameworks, diagramming and notational techniques within UML, visual nature of UML, CASE tools, structural models in UML; through the use of class diagrams, component diagrams and deployment diagrams, behavioural models in UML; through the use of use-case models, state-chart diagrams and activity diagrams.

UNIT IV OBJECT-ORIENTED DEVELOPMENT METHODOLOGY

Managing analysis and design, evaluation testing, coding, maintenance, metrics, importance of development methodology, object-oriented processes in the context of a development life cycle, Rational Unified Process (RUP), RUP splits project life cycle into four phases: inception, elaboration, construction and transition, workflows and models within each phase.

UNIT V REQUIREMENT AND ANALYSIS MODELLING

Requirements modeling, requirement model principle, use-case models, domain object models, documentation the requirements model, challenges and benefits of requirement modeling, agile requirement modeling, foundation class libraries, object-oriented databases, client/server computing, middleware, design vs. analysis, persistent objects, cohesion and coupling, analysis modeling, use-case analysis, analysis models.

References Books:

1. Object-Oriented Analysis and Design, John Deacon, Pearson Education, 2009.
2. Object-Oriented Modeling and Design, James Rumbaugh, Michael Blaha,William Premerlani, Prentice Hall, 1991.
3. Object-Oriented Design, Grady Booch, 1991.
4. Applying UML and Patterns, Craig Larman, Addison Wesley, 2000.
5. The Unified Modeling Language User Guide, Grady Booch, James Rumbaugh, Ivar Jacobson, Addison-Wesley Longman, 1999.
6. Object Oriented System Development, Ali Bahrami, McGraw Hill International Edition, 1999.
7. Analysis Patterns, Fowler, Addison Wesley, 1996.
8. Design Patterns, Erich Gamma, Addison Wesley, 1994.
9. Object-Oriented Systems Analysis and Design Using UML 2/e , Simon Bennett, Steve McRobb, Ray Farmer McGraw-Hill Education 2002.

FORMAL METHODS			
Course Code:	**CS405**	**Credits:**	**4**
No. of Lectures (Hrs/Week):	**3+1**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45+15**	**End Sem Exam Hours:**	**3**

UNIT I INTRODUCTION

Why formal methods were developed, problems in natural language specifications, formal versus informal programming, advantages of formal methods, requirements of formal system, types, prepositionallogic, predicatelogic, relationships and functions.

UNIT II FORMALSPECIFICATIONSTYLE

Model-oriented, specifications, concurrency-based specifications, example specification languages.

UNIT III VDM

Introduction to VDM, basic types, quote types, compound types, optional types, functions, operations, additional constructs, modules.

UNIT IV THEZNOTATION

Interchangelanguage,user-definedidentifiers,datatypes,basictypes, compound types, schemas, additional constructs.

UNIT V FORMALSEMANTICSANDTOOLS

Operational semantics, denotationa, semantics, axiomatic semantics proof editors, proof analyzer, symbolic simulators, translators, testg eneration tools.

Text Books:

1. AndrewHarry, "Formal Methods:FactFileVDMandZ",JohnWileyand Sons,1996.

Reference Books

2. Jim Woodcock, Jim Davies,"UsingZSpecification,RefinementandProof",Prentice Hall International, 1996.

RESEARCH TECHNIQUES IN ICT			
Course Code:	**CS527**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45**	**End Sem Exam Hours:**	**3**

UNIT I INTRODUCTION TO RESEARCH TECHNIQUES

Meaning of research, objectives of research, motivation in research, types of research (Introduction to experimental test bed, algorithmic research, simulation research, mathematical modeling approach), characteristics and prerequisites of research, significance of research, research process, sources of research problem, criteria of identifying the problem, necessity of defining the problem, errors in selecting research problem, technique involved in defining the problem, report and paper writing.

UNIT II DATA ANALYSIS ANDSTATISTICAL TECHNIQUES

Data and their analyses, quantitative methods and techniques, Measure of central tendency, measures of variation, frequency distribution, analysis of variance, methods, Correlation analysis, regression analysis, time series and forecasting, introduction to discriminant analysis, factor analysis, cluster analysis, conjoint analysis, probability distribution, binomial distribution, poisson distribution, uniform distribution, exponential distribution, and normal distribution, sampling methods, test of hypothesis.

UNIT III MATHEMATICAM MODELING

Steps of modeling, operations research models like queuing theory, stochastic processes, application of models, conceptual framework development and validation techniques, optimization techniques.

UNIT IV ALGORITHMIC RESEARCH

Algorithmic research problems, types of algorithmic research, types of solution procedure, steps of development of algorithm, steps of algorithmic research, design of experiments,

UNIT V SIMULATION ANDSOFT COMPUTING TECHNIQUES

Introduction to soft computing, artificial neural network, genetic algorithm, fuzzy logic and their applications, tools of soft computing, need for simulation, types of simulation, simulation language, fitting the problem to simulation study, simulation models, output analysis, data simulation packages like MATLAB, NS2, ANSYS, Cadence.

Reference Books:

1. Research Methodologies, R. Panneerselvam, Prentice Hall, 2007.
2. Research in Education, Best John V. and James V Kahn, Wiley eastern, 2005.
3. Elements of Educational Research, Sukhia, S.P., P.V. Mehrotra, and R.N. Mehrotra, PHI publication, 2003.
4. Methodology of Research Education, K. Setia, IEEE publication, 2004.
5. Research methodology, Methods and Techniques, Kothari, C.R., 2000.

COMPILER DESIGN LAB			
Course Code:	**CS481**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45**	**End Sem Exam Hours:**	**3**

1. Practice of Lex/Yacc of Compiler writing.
2. Write a program to check whether a string belongs to the grammar or not.
3. Write a program to generate a parse tree.
4. Write a program to find leading terminals.
5. Write a program to find trailing terminals.
6. Write a program to compute FIRST of non-terminals.
7. Write a program to compute FOLLOW of non-terminals.
8. Write a program to check whether a grammar is left recursive and remove left recursion.
9. Write a program to remove left factoring.
10. Write a program to check whether a grammar is Operator precedent.
11. To show all the operations of a stack.
12. To show various operations i.e read, write and modify in a text file
13. Write a program to implement Thomson's construct.

SOFTWARE ENGINEERING

(SEMESTER - VIII)

ADVANCED SOFTWARE ENGINEERING			
Course Code:	**CS532**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45**	**End Sem Exam Hours:**	**3**

UNIT I SOFTWARE ENGINEERING

Introduction to software engineering, Software Development Life Cycle, software process models, requirement analysis and design, software design process, coding, testing, implementation and maintenance, software metrics, agile software engineering, clean room software engineering, empirical software engineering.

UNIT II OBJECT-ORIENTED SOFTWARE ENGINEERING

OOSE, object-orientation paradigm, object-oriented programming languages, object modeling languages, object-oriented analysis, object-oriented design: design stereotypes and objects, design patterns; software design using patterns and framework, object-oriented case tools.

UNIT III COMPONENT-BASED SOFTWARE ENGINEERING

Component-Based Software Engineering (CBSE), CBSE and software reuse, CBSE vs. object-oriented software engineering, CBSE processes, domain engineering, component engineering, component-based software development life cycle, component vs. object, component-oriented programming, component-oriented programming vs. object-oriented programming, overview of component-based technology.

UNIT IV ASPECT-ORIENTED SOFTWARE ENGINEERING

Software engineering with aspects, aspects, aspect vs. object, aspect vs. component, join points and pointcuts, separation of concerns, crosscutting concerns, problems caused by scattering and tangling, system development, concepts of Aspect Oriented Programming, comparison to other programming paradigms.

UNIT V SOFTWARE RE-ENGINEERING AND REVERSE ENGINEERING

Re-engineering concept and approaches, redevelopment vs reengineering, reengineering process, reverse engineering, levels of reverse engineering: re-documentation, design recovery, specification recovery, conditions for reverse engineering, supporting techniques: forward engineering, restructuring, re-engineering, benefits of reverse engineering, software re-engineering patterns, patterns based software re-engineering, software re-engineering techniques.

Text Books:

1. Pankaj Jalote, An Integrated Approach to Software Engineering, Narosa Publishing House, New Delhi 1997.
2. Ian Sommerville, Software Engineering, Pearson Education, 2009.

Reference Books:

3. Pressman Roger S., Software Engineering: Practitioner's Approach, McGraw-Hill Inc., 2004.
4. N. S. Gill, Software Engineering: Software Reliability, Testing and Quality Assurance, Khanna Book Publishing Co (P) Ltd., New Delhi, 2002.
5. J. Rumbaugh, M. Blaha, W. Premerlani, Object-Oriented Modeling and Design, PHI, 1991.
6. George T. Heineman, William T. Councill, Component-Based Software Engineering: Putting the Pieces Together, Addision Wesley, 2001.
7. Robert E. Filman, Tzilla Elrad, Siobhán Clarke, Mehmet Aksit, Aspect-Oriented Software Development Addison-Wesley Professional, 2004.

OPEN SOURCE SOFTWARE SYSTEM			
Course Code:	**CS534**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45**	**End Sem Exam Hours:**	**3**

UNIT I OPEN SOURCE SOFTWARE

Open Source Software (OSS), history, philosophy and ethics of open source software, Pernes' principle, open source software development methodology, open source vs. closed source, open source software vs. free software, open source software vs. source available, Windows and Linux, open source development environment, methods and models, standards, open source standards, benefits of open standards, standard setting organizations and processes, project management via open source and open standard, OSS in e-government.

UNIT II OPEN SOURCE TECHNOLOGY

Open source technology and platform, Operating system: Linux, Berkeley Software Distribution; web server: Apache; communication servers: send mail, jabber; application and messaging server: JBoss, Zope, Zend.

UNIT III OPEN SOURCE LANGUAGES

Ruby, Ruby and object-orientation, data, expressions and flow control, class, object and modules, project and libraries, developing a basic Ruby application, PHP, configure environment, PHP basic, functions, arrays, object-oriented PHP, MYSQL, PostgreSQL.

UNIT IV OPEN SOURCE SOFTWARE APPLICATIONS AND FRAMEWORK

Open source desktop applications, Wiki software, LAMP application, web server and database server application, OSS management tools: taskjuggler, dotProject.net, rapid web application development framework: Ruby on Rail, Model-View-Controller model, Don't Repeat Yourself principle.

UNIT V OPEN SOURCE IN THE ENTERPRISE

Nature of open source, leadership in open source software life cycle, comparison in the risks of commercial and open source software, measuring the maturity of open source, designing an open source strategy, open source licenses, comparison of open source licenses, open source empowerment.

Text Books:

1. Paul Kavanagh, Open Source Software: Implementation and Management, Digital Press, 2004.
2. W. Jason Gilmore, Beginning PHP and MySQL, Apress, 2010.
3. Timothy Fisher, Ruby on Rail, Apress, 2009.

Reference Books:

4. Dan Woods, Open Source for the Enterprise: Managing Risks, Reaping Rewards, O'Reilly, 2005.
5. James Lee, Brent Ware,Open Source Web Development with LAMP, Pearson Education, 2008.
6. Steven Weber, The Success of Open Source, HarvardUniversity Press, 2004.
7. Peter Cooper, Beginning Ruby, Apress, 2007.

OBJECT-ORIENTED SOFTWARE ENGINEERING			
Course Code:	**CS536**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45**	**End Sem Exam Hours:**	**3**

UNIT I OBJECT-ORIENTED SOFTWARE ENGINEERING

Software engineering principles and concepts, software life cycle models, object-oriented software engineering, object-oriented concepts and paradigms, object-oriented methodology, object-oriented development process, object-oriented programming, object-oriented case tools, benefits and applications object-oriented programming.

UNIT II OBJECT-ORIENTED ANALYSIS AND DESIGN

Analysis and design, object-oriented analysis, static and dynamic analysis, object-oriented analysis process, object-oriented analysis techniques, software design, design methodology, object-oriented design concepts and design patterns, objects modeling language: Unified Modeling Language (UML).

UNIT III DESIGNING SOFTWARE USING PATTERNS

Process of design, principles, techniques, software architecture, architectural patterns, abstraction-occurrence pattern, hierarchy pattern, player-role pattern, singleton pattern, observe pattern, delegation pattern, adapter pattern, facade pattern, immutable pattern, read only interface pattern, proxy pattern.

UNIT IV OBJCET-ORIENTED METRICS

Metrics and measurement, metrics for object-oriented software development environments, characteristic of object-oriented metrics, Chidamber and Kemerer's metrics suite: Weighted Methods Per Class (WMC), Depth of Inheritance Tree (DIT), Number of Children (NOC), Coupling Between Object Classes (CBO), Response For a Class (RFC), Lack of Cohesion in Methods (LCOM), Lorenz and Kidds' metrics.

UNIT V DESIGN METRICS AND OBJCET-ORIENTED TESTING

Design metrics overview, method size, method internals, class size, class inheritance, method inheritance, class internals, MOOD (Metrics for Object-Oriented Design): Method Hiding Factor (MHF), Attribute Hiding Factor (AHF), Method Inheritance Factor (MIF), Attribute Inheritance Factor (AIF), Polymorphism Factor (PF), Coupling Factor (CF), object-oriented testing, test case design for object-oriented software, testing methods at class level: random testing of object-oriented class; interclass test case design: multiple class testing, test derived from behavior models.

Reference Books:

1. Object-Oriented Software Engineering, Bernd Bruegge, Allen H. Dutoit, PHI, 2003.
2. Object-Oriented Software Engineering, Timothy C. Lethbridge, Robert Laganiere, TMH, 2008.
3. Object-Oriented Modeling and Design, J. Rumbaugh, M. Blaha, W. Premerlani, PHI, 1991.
4. Object-Oriented Design, Grady Booch, 1991.
5. Software Engineering: Practitioner's Approach, Pressman Roger S., TMH, 2004.
6. Software Engineering: Software Reliability, Testing & Quality Assurance, N. S. Gill, KBP, 2002.
7. Fundamental of Software Engineering, Rajib Mall, Prentice Hall of India, 2003.
8. Software Engineering Concepts, Richard E. Fairley, Tata McGraw Hill, 1997.
9. An Integrated Approach to Software Engineering, Pankej Jalote, Narosa Publishing, 1997.
10. Software Engineering, Ian Sommerville, Pearson Education, 2009.

SOFTWARE ENGINEERING

(ELECTIVES - 1)

SOFTWARE ARCHITECTURE AND DESIGN			
Course Code:	**CS542/CS442**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45**	**End Sem Exam Hours:**	**3**

UNIT I SOFTWARE ARCHITECTURE

Foundations of software architecture, software life cycle architectural styles, quality attributes, architectural patterns, pipes and filters, layered systems, repositories, frameworks, patterns, methodologies, processes and process control, functional and non-functional properties of software architectures, heterogeneous architectures, virtual machine architecture, data flow architecture, service oriented architecture.

UNIT II DESIGN FUNDAMENTALS AND METHODOLOGIES

Nature of design process: objectives, building modules, constructs, design qualities, assessing the design, design viewpoints for software, design strategies: top down and bottom up, organizational methods and design, Jackson structural programming, Jackson system development, models for software architecture

UNIT III SOFTWARE ARCHITECTURE DESIGN

Architectural design and mapping, architecture design patterns, module architecture view,styles of the module view type, execution architecture view, code architecture view, component-and-connector viewtype, styles of component-and-connector viewtype, allocation viewtype and styles, object-oriented architecture, user interface architecture, quantified design space, formalizing architectural description language, first class connectors, tools for architectural design: Unicon, A4; exploiting style in architectural design, architectural interconnection.

UNIT IV INTERACTION ORIENTED SOFTWARE ARCHITECTURE and Design

Model-View-Controller (MVC), Presentation-Abstraction-Control (PAC) architecture, distributed architecture: client server architecture, multi-tire, service-oriented architecture (SOA). Design principles, traditional approach to design, Structured Analysis Design Technique (SADT), Structures System Analysis and Design Method (SSADM), user interface design; human factor, human computer interaction, interface design guide lines, standards, object-oriented analysis and design.

UNIT V PATTERNS

Design patterns, creational patterns, access control patterns, service variation patterns, service extension patterns, archetypes patterns, model driven architecture with archetype patterns, literate modeling, Customer Relationship Management (CRM) archetype pattern, product archetype pattern, quantity archetype pattern, rule archetype pattern, layering, organizing domain logic, mapping to relational databases, web presentation, domain logic patterns, data source architectural patterns, object-relational behavioral patterns, object-relational structural patterns, object-relational metadata mapping patterns, web presentation patterns, distribution patterns, offline concurrency patterns.

Text Books:

1. Software Architecture Perspectives on an Emerging Discipline, M. Shaw Prentice-Hall, 1996.
2. Software Architecture Design: Methodology and Styles, Lixin Tao, Xiang Fu and Kai Qian, Stipes Publishing L.L.C., 2006.
3. Software Architecture in Practice, Len Bass, Paul Clements, Rick Kazman, Pearson Education Asia, 2003.

References Books:

4. Software Design, David Budgen, Addison-Wesley, 1994.
5. Software Engineering, Pressman R.S, McGraw Hill Inc., 1996.
6. Structured System Analysis and Design methods Application and Context, Ed Downs, Peter Clare, Jan Coe, Prentice Hall, 1998.
7. Design Patterns for Object-Oriented Software Development, Wolfgang Pree, Addison- Wesley, 1995.

8. Software Architecture Resource website.
   http://www2.umassd.edu/SECenter/SAResources.html.
9. Essential Software Architecture, Ian Gorton Springer, 2006.
10. Pattern-Oriented Software Architecture, Frank Buschmann, Hans Rohnert, Kevin Henney, Douglas C. Schmidt, Wiley, 2004.

SOFTWARE MEASUREMENT AND ESTIMATION			
Course Code:	**CS544/CS444**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45**	**End Sem Exam Hours:**	**3**

UNIT I SOFTWARE MEASUREMENTS THEORY

Fundamentals of software measurement, measurement scale, scope of software, need for measurement, type of measurement process, measures of central tendency and variability, validation, validity and reliability of measurement, empirical investigation, planning experiments, challenges of software measurement, measurement models, data collection, analysis methods, statistical methods, measurement life cycle.

UNIT II MEASURING SOFTWARE SYSTEM

Measuring size of software, physical measurement of software, measuring functionality, measuring complexity, structural complexity, conceptual complexity, estimating efforts.

UNITIII MEASUREMENT AND METRICS

Software metrics, Design metrics: method size, method internals, class size, class inheritance, method inheritance, class internals, class externals; software quality metrics: product quality, process quality, metrics for software, maintenance; metrics for reliability prediction, measuring the costs of defect removal, evaluating defect prevention methods, Rayleigh Model, Problem Tracking Report (PTR) model, reliability growth model, model evaluation, orthogonal defect classification, case studies of metrics programs.

UNIT IV SOFTWARE QUALITY MODELS AND RELIABILITY

Software quality models: Boehm's model, McCall's model, ISO 9126 model, Basic software quality metrics, Quality management models, Measuring customer satisfaction , Software Quality Assurance (SQA), Defects, Faults vs. failures, Defect Projection Techniques and Models, Software Reliability Measurement and Prediction, The Cost of Reliability, Software Reliability Theory, reliability models, failure arrival rate, response time, response time measurements, availability measuring progress, case studies of software quality.

UNIT V SOFTWARE COST ESTIMATION

Software estimation methodologies and models, combining estimates, estimating Issues, software cost factors, cost estimation, software cost estimation techniques, staffing-level estimation, estimating software maintenance costs, Cost estimation Constructive Cost Model (COCOMO), Function Point (FP) model, Common Software Measurement International Consortium (COSMICS), Full Function point (FFP) approach, software estimation crisis, case studies

References Books:

1. Software Measurement and Estimation: A Practical Approach, Linda M. Laird, M. Carol Brennan, Willy, 2006.
2. Software Metrics, N. E. Fentar and S. L. Pflieger, International Thomson Computer Press, 1997.
3. Metric and Models in Software Quality Engineering, Stephen H. Kin, Addision Wesley, 1995.
4. Measuring Software Process, William A. Florac and Areitor D. Carletow, Addison - Wesley, 1995.
5. Estimating Software Costs: Bringing Realism to Estimating, Capers Jones, Tata M. Hill, 2007.
6. Applied Software Measurement: Global Analysis of Productivity and Quality, Capers Jones, Tata McGRAW Hill, 2008.

SOFTWARE RE-ENGINEERING			
Course Code:	**CS546/CS446**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45**	**End Sem Exam Hours:**	**3**

UNIT I INTRODUCTION TO RE-ENGINEERING

Re-engineering concept and approaches, growing problems and maintaining software, redevelopment vs reengineering, reengineering process and methods, Reengineering economics, BPR process, Lehman's law, Pitfalls of reengineering, Technology for reengineering.

UNIT II REVERSE ENGINEERING

Function abstraction, data abstraction, Process abstraction, levels of reverse engineering: re-documentation, design recovery, specification recovery, conditions for reverse engineering, supporting techniques: forward engineering, restructuring, re-engineering, benefits of reverse engineering.

UNIT III SOURCE CODE TRANSLATION ANDDATA REENGINEERING

Need ofsource code translation: hardware platform update, organizational policy change, lack of software support, understandable software code, testing and maintainable, detection of duplicate code. Data reengineering and Migration, Documentation.

UNIT IV HYBRID RE-ENGINEERING

Hybrid re-engineering tracks: translation of existing code, Commercial of the Shelf (COTS), custom code, hybrid re-engineering approach, risks of hybrid re-engineering: schedule, functionality, cost, quality, interface and interoperability, and benefits of hybrid re-engineering.

UNIT V SOFTWARE RE-ENGINEERING PATTERNS AND TECHNIQUES

Software re-engineering patterns, patterns based software re-engineering, object-oriented re-engineering patterns & technique, design patterns, testing patterns, software re-engineering techniques: restructuring, refactoring and data re-engineering, forward re-engineering, Clean room approach, tools support for re-engineering.

Reference Books:

1. Software Re-engineering, Robert S. Arnold, IEEE Computer Society Press, 1993.
2. Reversing: Secrets of Reverse Engineering, Eldad Eilam, John Wiley and Sons, 2005.
3. Software Engineering, James F. Peters, Witold Pedrycz, Willey, 2008.
4. Object-Oriented Reengineering Patterns, Serge Demeyer, Morgan Kaufmann, 2003.
5. Introduction to Software Engineering, Ronald Leach, CRC, 1999.
6. Component-Based Software Engineering: Alan W. Brown, Wiley-IEEE Computer Society, 1996.
7. Software Engineering: A Practitioner's Approach, Pressman, Roger, McGraw Hill, 1997.

SOFTWARE REUSABILITY			
Course Code:	**CS548/CS448**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45**	**End Sem Exam Hours:**	**3**

UNIT I REUSABILITY

Motivation for reuse, reuse data element, reuse requirements, reuse activities, reuse design, reuse source code, reuse interface, reuse development plan, reuse test plan, reuse test case, reuse driven organizations, design patterns, generators based reuse, application framework, managing a reuse project, characteristics of reuse project, adapting a project to reuse, reuse tools, cost effective techniques for reuse, composition-based and generation-based reuse techniques.

UNIT II APPLICATIONS AND COMPONENT SYSTEM

Reuse object-oriented software engineering model, component applications, facades control access to component systems, facades and component systems, components specialization for reuse, variability and its mechanism, reuse of variable components to build application systems.

UNIT III OBJECT COMPONENTS

Object models, reusing analysis and design components, expressing variability in object model components, subsystem components group related classes, reusable design and implementation, components packaging and documenting object components. Reuse development processes, develop for reuse, develop with reuse, testing of reusable components, object-oriented components techniques and life cycles, object-oriented development for reuse, detailed design for reuse, implementation for reuse, Verification and Validation.

UNIT IV COMOPONENT SYSTEM ENGINEERING

Building flexible component systems, analyzing requirements focusing on variability, performing robustness analysis, designing and testing the component system, packaging of component system for reuse, principles of component design and reuse, design prototyping, Component-Based Software Reuse (CBSR), component develop for reuse, develop with reuse, testing of reusable component, reuse metrics, challenges in CBSR.

UNIT V APPLICATION SYSTEM ENGINEERING

Application system reuse, building application systems from reusable components, analyzing requirements, performing robustness analysis for flexible application systems, designing, implementing, testing the application system, packaging application system.

Reference Books:

1. Software Reuse: Architecture, Process and Organization for Business Success, Ivar Jacobson, Martin Griss, Patrik Jonsson, Pearson Education, 2009.
2. Software Reuse: A Hoilstic Approach, Even-Andre Karisson, John Wiley and Sons, 1996.
3. Software Reuse Techniques: Additional Reuse to the Systems Development Process, Karma McClure, Prentice Hall, 1997.
4. Software Reusability, Wilhelm Schafer, Ruben Prieto-Diaz, Masao Matsumoto, Prentice Hall, 1993.
5. Component-Based Software Engineering: Alan W. Brown, Wiley-IEEE Computer Society, 1996.

WEB-BASED SOFTWARE ENGINEERING			
Course Code:	**CS550/CS450**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45**	**End Sem Exam Hours:**	**3**

UNIT I

Introduction, analysis, architectural design, design patterns, formulation, interface design, navigation, design, project, management, quality, attributes, structures, testing, WebApp, attributes, WebApp categories, WebE process, WebE team.

UNIT II

Attributes of web based applications, quality attributes, technologies, web application quality, the web process, framework for the web, web process model

UNITIII

Formulating and analyzing web based system, design for web based applications, testing web based applications, management issue, project management.

UNIT IV

Reviews general architectures for web application and technology-aware application designs, discusses the concepts and techniques for engineering and evaluating user interfaces appropriate for a web application's intended audience.

UNIT V

Explores the interaction between users and the application's user interface, special attention will be paid to web technologies and standards available for audiences with special needs.

Related research papers reading as suggested by subject Teacher and their analysis.

Reference Books:

1. Web Engineering: A Practitioner's Approach by Pressman and Lowewhich considers the Web engineering process in its entirety.
2. Web Engineering: Principles and Techniques [Paperback] by Woojong Suh
3. Roger Pressman.S., " Software Engineering : A Practitioner's Approach ", (3rd Edition), McGraw Hill,

OPEN SOURCE SOFTWARE SYSTEM LAB			
Course Code:	**CS586**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**		
Total No. of Lectures:	**10**	**End Sem Exam Hours:**	**3**

Experiments

1. Study, analysis and compare various open source software according to the open source standards.
2. Download any open source software application written in JAVA with source code and modify the source code to add extra functionality.
3. Develop a small application with the help of database (MySQL and PostGre SQL) for any domain in JAVA technology.
4. Learn the working and installation of the following tools:
   Jboss, Zope, Zend.
5. Learn the working and installation process of the following Operating Systems and deliver a presentation: Linux, Fedora, Ubuntu, CentOS and Boss.
6. Learn the working and installation process of the following Open Source Software Management Tools : Taskjuggler and dotProject.net
7. Introduction to MySQL, an exercise on data types in MySQL.
8. Exercise on Data Definition Language, Data Manipulation Language and transaction control using MySQL.
9. Exercise on Types of Data Constraints using MySQL and Exercise on different types of sub queries using MySQL.
10. Introduction to PL/SQL, Control Structures, Procedures and Functions, View using MySQL.

SOFTWARE ENGINEERING

(SEMESTER - IX)

SOFTWARE TESTING			
Course Code:	**CS631**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45**	**End Sem Exam Hours:**	**3**

UNIT I SOFTWARE TESTING

Essentials of software testing, testing methodology, testing and debugging, software and hardware testing, verification and validation, correctness vs. reliability, challenges in software testing, software testing life cycle (STLC), flow graphs and path testing, transaction flow testing, data flow testing, database testing, web-based testing.

UNIT II SOFTWARE TESTING TECHNIQUES

testing levels: unit testing, integration testing, system testing, acceptance testing, testing techniques: white box testing, black box testing; thread testing, regression testing, alpha testing, beta testing, static testing, dynamic testing, performance testing, ad hoc testing, smoke testing,exhaustive testing, structural testing, mutation testing; Testing Maturity Model (TMM), verification process, defect tracking, severity and priority, defects, fault, failure, bug, bug life cycle, bug report and bug reporting tools.

UNIT III TEST METRICS AND MEASUREMENT

Purpose of test plan, test plan design, test script, test cases, test management, test case specification, executing test cases, test result analysis. Metrics and measurement, project metrics: effort variance , schedule variance, effort distribution across phases, progress metrics: test defect metrics, development defect metrics, productivity metrics, defect density, defect leakage ratio, Residual Defect Density (RDD), test phase effectiveness, test reports.

UNIT IV SOFTWARE VERIFICATION AND VALIDATION

Verification, methods of verification, types of review on the basis of stage, reviews in testing life cycle, coverage in verification, concerns of verification, validation, coverage in validation, management of Verification and Validation (V & V), software development V & V activities.

UNIT V SOFTWARE TESTING TOOLS

Manual vs. automated testing, functionality and regression testing tool: Win Runner ; load and performance testing tool: Load Runner; web based application testing tool: Quick Test Professional (QTP); Rational Seed Tool for requirement analysis to testing and project management.

Reference Books:

1. Effective Methods for Software Testing, William E. Perry, John Wiley and Sons, 2002.
2. Effective Software Testing: 50 Specific Ways to Improve Your Testing, Dustin, Pearson Education, 2002.
3. An Integrated Approach to Software Engineering, Pankej Jalote, Narosa Publishing House, New Delhi 1997.
4. The Art of Software Testing, Glenford J.Myers, John Wiley & Sons, 1979.
5. Software Testing: A Craftman's Approach, P. C. Jorgensen, CRC Press, 1995.
6. Software Testing Techniques, Boris Beizer, Dreamtech, 2006.
7. Software Testing: Principles and Practices, Srinivasan Desikan, Gopalaswamy Ramesh, Pearson Education, 2008.
8. Software Testing, Aditya P. Mathur, Pearson Education, 2008.
9. Software Testing: Principle, Techniques and Tools, M. G. Limaye, Tata McGraw Hill, 2009.

COMPONENT-BASED SOFTWARE ENGINEERING			
Course Code:	**CS633**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45**	**End Sem Exam Hours:**	**3**

UNIT I INTRODUCTION TO CBSE

Component-Based Software Engineering (CBSE), CBSE vs. Object-Oriented Software Engineering, CBSE methodology, CBSE processes, domain engineering, component engineering, component-based software life cycle, component vs. object, CBSE project management, measurement and metrics for CBSE, challenge CBSE, advantages and disadvantages of CBSE, economics of CBSE.

UNIT II COMPONENT-ORIENTED PROGRAMMING

Component-oriented programming, object-oriented programming to component-oriented programming, component-oriented programming vs. object-oriented programming, principle and infrastructure of component-oriented programming.

UNIT III COMPONENT AND COMPONENT MODEL

Component, component technology, software component, specification of software component, component architecture, component framework, component interface, component abstraction, component services, components model, component selection, component adaptability, component certification, component composition, component and interface modeling, domain modeling, patterns and frameworks.

UNIT IV COMPONENT-BASED DESIGN AND REUSE

Principles of component design and reuse, design prototyping, design production, design refactoring, design documentation, component-based software reuse, reusable component, component-based reuse metrics.

UNIT V COMPONENT TECHNOLOGIES

Component technologies: Component Object Model (COM), Distributed Component Object Model (DCOM), Common Object Requesting Broker Architecture (CORBA), Enterprise Java Beans (EJB).

Text Books:

1. George T. Heineman, William T. Councill, Component-Based Software Engineering: Putting the Pieces Together, Addision Wesley, 2001.
2. Andy Ju An Wang, Kai Qian, Component-Oriented Programming, Willey Interscience, 2005

Reference Books:

3. Clemens Szyperski, Component Software: Beyond Object-Oriented Programming, Addison Wesley, 1997.
4. Alan W. Brown, Component-Based Software Engineering, Wiley-IEEE Computer Society, 1996.
5. Sudha Sadasivam, Component-Based Technology, G. Willy, 2008.
6. Pressman Roger S., Software Engineering: Practitioner's Approach, McGraw-Hill Inc., 2004.
7. N. S. Gill, Software Engineering: Software Reliability, Testing and Quality Assurance, Khanna Book Publishing Co. (P) Ltd., New Delhi, 2002.

SOFTWARE TESTING LAB			
Course Code:	**CS681**	**Credits:**	**2**
No. of Lectures (Hrs/Week):	**3**		
Total No. of Lectures:	**10**	**End Sem Exam Hours:**	**2**

Programs/Experiments List:

1. Introduction to various phases of SDLC Implementation.
2. Create the Requirement and Design document using IEEE format.
3. Design and Manage the Test Plan using Rational Test Manager.
4. Write a program to design the Test Plan.
5. Write a program to calculate the Cyclomatic Complexity.
6. Design Test Cases using Rational Test Manager.
7. Develop and Execute Manual Test.
8. Implement Data-Driven Testing.
9. Create and modify various types of Verification points.
10. Determine Test Results and list out the Test Case Distribution Report.

SOFTWARE ENGINEERING

(ELECTIVES - 2&3)

ASPECT-ORIENTED SOFTWARE ENGINEERING			
Course Code:	**CS641/C545**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45**	**End Sem Exam Hours:**	**3**

UNIT I ASPECT-ORIENTED SOFTWARE ENGINEERING

Software engineering with aspects, aspect-oriented software evolution, aspects, aspect vs. object, aspect vs. component, join points and pointcuts, separation of concerns, crosscutting concerns, problems caused by scattering and tangling.

UNIT II ASPECT-ORIENTED PROGRAMMING

Concepts of AOP, inter-type declarations, implementation, comparison to other programming paradigms, nature of aspect-orientation, concepts and terminology, Join Point Model, AspectJ Point Model, pointcut designators, inter-type declarations, aspect weaving, comparison with object-oriented programming.

UNIT III ASPECT-ORIENTED REQUIREMENT ENGINEERING

Aspect-oriented requirements engineering and process, aspect-oriented requirements notations, aspect-oriented requirements tool support, adoption and integration of aspect-oriented requirements engineering, and assessment/evaluation of aspect-oriented requirements.

UNIT IV ASPECT-ORIENTED SOFTWARE ARCHITECTURE

Aspect-oriented software architecture and process, aspect-oriented architecture notations, aspect-oriented architecture tool support, adoption and integration of aspect-oriented architecture, and assessment/evaluation of aspect-oriented architecture.

UNIT V ASPECT-ORIENTED MODELING AND DESIGN

Aspect-Oriented Modeling, AOM approach, aspect model, aspect-oriented design, aspect-oriented design process, aspect-oriented design notations, aspect-oriented design tool support, aspect-oriented design, AspectJ, Aspect Werkz, Hyper/J, Java Aspect Component.

Reference Books:

1. Aspect-Oriented Software Development, Robert E. Filman, Tzilla Elrad, Siobhán Clarke, Mehmet Aksit, Addison-Wesley Professional, 2004.
2. Aspect-Oriented Software Development with Use Cases, Ivar Jacobson, Addison-Wesley Object Technology Series, 2005.
3. Aspect-Oriented Analysis and Design: The Theme Approach, Siobhán Clarke, Addison-Wesley Object Technology Series, 2005.

SOFTWARE RELIABILITY			
Course Code:	**CS643/CS547**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45**	**End Sem Exam Hours:**	**3**

UNIT I SOFTWARE RELIABILITY

Measures of software reliability, Mean Time To Failure (MTTF), Mean Time Between Failure (MTBF), Mean Time To Recovery (MTTR), availability, maintainability, Musa's operational profiles and type-1 uncertainty, defect removal and type-2 uncertainty, reliability stability and reliability growth, hardware reliability vs. software reliability, failure probability density function and reliability function, Reliability prediction , reliability metrics.

UNIT II DEVELOPMENT OF RELIABLE SOFTWARE

Reliable software, defect prevention, detection and removal, design for robustness, verification & validation, stabilization of requirements, design, code and test artifacts, active and passive fault detection, fault handling and correction, exceptions, survivability, reliability models, software availability model.

UNIT III FAULT TOLERANCEIN HARDWARE SYSTEMS

Fault classification, fault tolerance attributes and system structure, fault prevention, anticipated and unanticipated fault, test generation for digital systems, combinational logic network, Boolean difference method, test generation for sequential circuits, fault simulation, application of hardware fault tolerance in developing fault tolerant software systems.

UNIT IV SOFTWARE AND HARDWARE FAULT TOLERANCE

Software and hardware faults, failure intensity function, characterization of fault injection, detection and correction, techniques for prediction of remaining faults and fault injection, classification tree analysis, code coverage, coding technique, fault tolerant & self checking, fail safe circuits, synchronous and asynchronous fail safe circuits.

UNIT V FAULT TOLERANT SOFTWARE

Concept of N-version programming (NVP) and methods, recovery block, acceptance tests, fault trees, validation of fault tolerant systems, security, fault tolerance in wireless/mobile networks and Internet.

Reference Books:

1. Software Reliability Engineering, John D. Musa, Tata McGRAW Hill, 2005.
2. Fault-Tolerant Computer System Design, D.K. Pradhan, 2003.
3. Design and Analysis of Fault-Tolerant Digital Systems, B. W. Johnson, Addison-Wesley, 1989.
4. Fault-Tolerant Computing, Theory & Techniques, D.K. Pradhan, Prentice Hall, 1986.
5. Reliable Computer Systems: Design and Evaluation, D. P. Siewiorek and R. S. Swartz, Digital Press, 1992.
6. Probability and Statistics with Reliability, Queueing and Computer Science application, K.S.Trivedi, Prentice Hall, 1982.
7. Fault Tolerant Principles and Practice, Anderson and Lee, PHI, 1989.

SOFTWARE QUALITY ASSURANCE			
Course Code:	**CS645/CS549**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45**	**End Sem Exam Hours:**	**3**

UNIT I SOFTWARE QUALITY AND ENGINEERING

Quality concepts and productivity relationship, software quality factors, software quality costs, Total Quality Management (TQM), continuous improvement cycle: Plan, Do, Check and Act (PDCA), quality policy, cost of quality, quality engineering, quality planning: goal setting and strategy formation, assessment and improvement.

UNIT II SOFTWARE QUALITY ASSURANCE (SQA)

Components of SQA, classification, defect detection, defect prevention, defect reduction, defect containment, QA activities in software processes, verification and validation, software review, inspection, formal verification, statistical software quality approach.

UNIT III COMPONENTS MEASUREMENT WITH REFERENCE TO SQA

Metrics, product quality metrics, process quality metrics, metrics for software maintenance, quality tools for quality control, test management and organizational structures, Capability Maturity Model (CMM), Capability Maturity Model Integration (CMMI), ISO 9000, quality and quality management metrics, Deming's Principle, SQA team formation

UNIT IV QUALITY MANAGEMENT MODEL

Integrating quality activities in project life cycle, reviews, software testing, strategies and implementation, Computer-Aided Software Engineering (CASE) tools, The Rayleigh model framework, code integration pattern, Problem Tracking Report (PTR), reliability growth model, Service Quality, Kano Model, Customer retention, continuous process improvement, Juran's Trilogy, TQM principles, Kaizen Technique, Statistical Quality Assurance, Mc call quality factors

UNIT V SOFTWARE QUALITY ASSURANCE BEYOND TESTING

Defect prevention and process improvement, root cause analysis for defect prevention, software inspection, inspection related activities, fault tolerance and failure containment, comparing quality assurance techniques and activities.

Reference Books:

1. Metrics and Models in Software Quality Engineering, Stephan H. Kan, Pearson Education, 2007.
2. An Integrated Approach to Software Engineering, Pankej Jalote, Narosa Publishing House, New Delhi 1997.
3. Making Sense of Software Quality Assurance, Raghav J. Nandyal, Tata McGRAW Hill, 2007.
4. Software Quality Assurance: A Practitioner Approach, Kaman Malik, Praveen Chaudhary, Tata McGRAW Hill, 2008.

SOFTWARE MAINTENANCE			
Course Code:	**CS647/CS553**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45**	**End Sem Exam Hours:**	**3**

UNIT I INTRODUCTION TO SOFTWARE MAINTENANCE

Evolution of software products, economics of maintenance, characteristics of software maintenance, product development life cycle, different type of software products, deployment model, adaptive maintenance, enhancement request, proactive defect prevention, maintenance process, problem reporting, problem resolution, fix distribution, software maintenance from customers' perspective, category of software maintenance.

UNIT II REVERSE ENGINEERING

Function abstraction, data abstraction, and process abstraction, levels of reverse engineering: re-documentation, design recovery, specification recovery, conditions for reverse engineering, supporting techniques: forward engineering, restructuring, re-engineering, benefits of reverse engineering.

UNIT III CONFIGURATION MANAGEMENT

Software configuration management process, patches, configuration management in global development teams, baseline, software configuration items, identification of objects in software configuration, version control, change control, configuration audit, status reporting, software configuration management standards, metrics for maintenance in configuration management. traditional process model: Code and Fix Model, Waterfall Model, Spiral Model; maintenance process model: Quick Fix Model, Boehm's Model, Osbornes' Model, Iterative Enhancement Model; process maturity model: Capability Maturity Model (CMM), Capability Maturity Model Integration (CMMI).

UNIT IV MAINTENANCE AND OTHER LIFE CYCLE ACTIVITIES

Design and maintenance, programming & maintenance, debugging and maintenance, testing and maintenance, maintenance management, maintenance management functions: planning, organizing, staffing, leading, controlling; maintenance management organizations: functional organization, project organization, matrix organization.

UNIT V MAINTENANCE MEASURES

Importance of integrity in measurement, software measure and metrics, objective of software measurement: evaluation, control, assessment, improvement, prediction, maintenance measures: size, complexity; quality: product and process quality, understandability and maintainability, impact analysis in creating maintainable system.

Reference Books:

1. Software Maintenance: Concept and Practice, Penny Grubb, Aramstrong A. Takang, International Thompson Publishing Inc., 1996.
2. Software Maintenance, Gopalaswamy Ramesh, Ramesh Bhattiprolu, Tata McGraw Hill, 2009.
3. Software Engineering: Software Reliability, Testing and Quality Assurance, Nasib S. Gill, Khanna Book Publishing Co (P) Ltd., New Delhi, 2002.
4. Software Engineering: Practitioner's Approach, Pressman Roger S., McGraw-Hill Inc., 2004.
5. Software Quality Engineering: Testing, Quality Assurance, and Quantifiable Improvement, Jeff Tian, John Wiley and Sons Inc., and IEEE Computer Society Press, 2005.
6. Software Quality Assurance, Daniel Gain, Pearson Education, 2009.

SOFTWARE PERFORMANCE			
Course Code:	**CS649/CS555**	**Credits:**	**3**
No. of Lectures (Hrs/Week):	**3**	**Mid Sem Exam Hours:**	**2**
Total No. of Lectures:	**45**	**End Sem Exam Hours:**	**3**

UNIT I SOFTWARE PERFORMANCE

Software and performance, repository and scalability, importance of performance, consequences and cause of performance failure, reactive and proactive performance management, software performance Engineering (SPE), modeling strategy and models, SPE for object-oriented system.

UNIT II SOFTWARE PERFORMANCE ENGINEERING PROCESS

SPE Process for object oriented system, performance solutions, principles, patterns, implementation solutions,

UNIT III SOFTWARE EXECUTION MODELS

Software Execution Models (SEM), representing of SEM, execution graphs and graph restrictions, model solution, basic solution algorithms, analysis procedure, execution graph from sequence diagrams.

UNIT IV SYSTEM EXECUTION MODELS

System Execution Models, system model basics, performance metrics, system models for SPE, advanced system models, Schedulability.

UNIT V SOFTWARE PERFORMANCE ENGINEERING

SPE Data Collection, SPE data requirement, key performance scenarios, performance objectives, execution environment software resource requirements, data gathering issues, performance walkthrough, resource estimation techniques.

Reference Books:

1. Performance Solution : A Practical Guide to Creating Responsive, Scalable Software, Addison-Wesley Professional.
2. Performance prototyping: a simulation methodology for software performance engineering, http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=218462&tag=1