1. Description of Courses (Session 2015 and Subsequent Sessions)

CS 101: Computing Fundamentals 3(2,1)

Introduction to Hardware, Software Engineering, Networking, Operating System and Database Concepts. Business Applications: Data acquisition, storage and presentation using MS Office. Pseudo Code, Problem solving techniques and Flow Charts Introduction to Visual Basic. Class Project: Solving Specific Engineering problem. Good understanding of the world wide and internet applications.

Texts:

1. *Computer Science an Overview* by J. Glenn Brook Shear.
2. *Simple Program Design* by Author: Lesley Anne Robertson.
3. Schaum Series' *Visual Basic and tutorials*.
4. *Discovering Computers* by Shelly Vermaat

CSE 121: Circuit Analysis-I 4(3,1)

Electric quantities, electric signals, electric circuits, Kirchhoff's laws, circuit elements. Resistance, series parallel combination, voltage and current dividers, resistive bridges and ladders, practical sources and loading, instrumentation and measurement. Nodal analysis, loop analysis, linearity and superposition, source transformation, One ports, circuit theorems,power calculations. Dependent sources, circuit analysis with dependent sources, ideal transformer, amplifiers. The operational amplifier, basic op-amp configurations, ideal op-amp circuit analysis, summing and difference amplifiers, amplifier types. Capacitance, inductance, natural response of RC and RL circuits. Response to DC forcing function. Transient response of first order circuits, step, pulse and pulse train responses, first order op-amp circuits. Transient response and step response of second order circuits.

Text: *Electric Circuit Fundamentals* by Sergio Franco (Latest ed.)

CS 141: Introduction to Computing 4(3,1)

This broad-based, entry-level course provides a general overview of the main concepts in computing, Topics studied include the history and evolution of computers, central processing unit, data storage, input/output devices, multimedia, operating systems, programming languages, networking, the Internet, systems analysis and design, management information systems, electronic commerce, security and privacy issues, ethical issues and the computing profession. Students will become familiar with popular operating systems, software applications and tools, and learn the basic concepts of programming by developing interactive web pages.

Text: *Computers: Tools for an Information Age*, 8th ed., H.L. Capron, Addison Wesley, 2003.

CS 142: Programming Fundamentals 4(3,1)

A rigorous introduction to programming and computer science with a substantial laboratory component. The course introduces students to at least three different styles in programming: functional, imperative, and object-oriented. Key topics include: recursive and iterative processes, procedural abstraction, substitution model, orders of growth, data abstraction, introduction to lambda-calculus, state and mutation, environment model of evaluation, object-oriented programming, interpreters, compilers.

Text: *Structure and Interpretation of Computer Programs* by Abelson, Sussman and Sussman (second edition, MIT Press, 1996)

CS 201: Numerical Analysis 4(3,1)

This course makes the students understand the basic techniques of numerical analysis for solving nonlinear algebraic equations, interpolation, for numerical differentiation and integration. It enables them to set up computational algorithms for the solution of above problems.

Text: *A First Course in Numerical Analysis*, Saeed Bhatti & Naeem Bhatti, 4th Edition, 2003.

CS 211: Discrete Mathematical Structures 3(3,0)

This course offers an intensive introduction to discrete mathematics as it is used in computer science. Topics include functions, relations, sets, simple proof techniques, boolean algebra, propositional logic, digital logic, elementary number theory, fundamentals of counting, predicate logic, recurrence relations, graphs, trees, matrices, computational complexity, elementary computability, and discrete probability.

Text: *Discrete Mathematics and its Applications*, Kenneth H. Rosen, 5th Edition, 2003, McGraw-Hill.

CS 212: Data Structures and Algorithms 4(3,1)

This course covers basic data structures and provides an introduction to algorithms. The following topics are covered in the course: array, list, stack, queues, basic algorithm analysis, trees, hash table, priority queues, sorting algorithms, and graphs. The course assignments reinforce programming skills of the students.

Text: *Data Structures and Algorithm Analysis*, M. A. Weiss, Pearson, 2nd Edition, 1997.

CSE 221: Digital Logic Design 4(3,1)

Design concepts. Introduction to logic circuits, truth tables, logic gates and networks, boolean algebra, synthesis using AND, OR and NOT gates, NAND and NOR logic networks, design examples, introduction to CAD tools, introduction to Verilog. Implementation technology. Optimized implementation of logic functions. Number representation and arithmetic circuits. Combinational circuit building blocks. Flip flops, registers, counters and simple processor.

Text: *Fundamentals of Digital Logic with Verilog Design* by Brown and Vranesic (Latest ed.)

CSE 222: Computer Organization and Assembly Language 4(3,1)

This course will provide an overview of the structure and function of computers at the register level and how data is represented at this level. Main topics of the course are: organization of computer hardware, the functions of assembler, linker, and loader, basic assembly language instruction set, memory management, addressing modes, stack and procedures, low-level I/O. Students will learn to design, implement, and debug programs in assembly language.

Texts:

1. *Assembly Language Programming and Organization of IBM PC* by Ytha Yu and Charles Marut, McGraw- Hill, 1992
2. *Computer Organization and Design,* David A. Patterson, John L. Hennessy, 3rd Edition.

CSE 223: Circuit Analysis - II 3(3,0)

Sinusoids and phasors, AC response of basic elements, first order circuits and second order circuits. Phasor algebra, AC impedance, Frequency domain analysis, op-amp AC circuits. AC power and 3-phase systems. AC Resonance. Laplace transform and its application to circuit analysis. Fourier Series analysis. Network functions, Bode plots.
Text: *Electric Circuit Fundamentals* by Sergio Franco (Latest ed.)

CS 241: Object-Oriented Programming 4(3,1)

This course is designed to teach object-oriented programming techniques to those who have learned basic programming concepts. Topics include: Classes; Objects; Data Abstraction; Data Encapsulation; Polymorphism; and Inheritance; Overloaded Operators; Overridden Methods; Friend Functions; Virtual Functions; Templates; and Files. C++ programming language is used to demonstrate the object-oriented programming constructs.
Text: *C++ How to Program*, Deitel & Deitel,4th Edition, Prentice Hall
C++ Programmer's Notebook, Jim Keogh, 2nd Edition.

CS 281: Object Oriented Analysis and Design 3(3,0)

This course requires prior experience in object-oriented programming and familiarity with basic object-oriented concepts such as abstraction, encapsulation, inheritance, composition and aggregation. Object-oriented analysis and design concepts are introduced using UML, and an iterative evolutionary design process based on the Unified Process applied in an agile spirit. Students will also learn applications of object-oriented design principles and patterns such as GRASP and GoF design patterns.
Text: *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development,* Craig Larman, 3rd edition, Prentice Hall, 2005 (or Latest edition).

CSE 301: Probability and Random Variables 3(3,0)

Axiomatic definition of probability spaces. Combinatorial methods. Conditional probability; product spaces. Random variables; distribution and density functions; multivariate distribution; conditional distributions and densities; independent random variables. Functions of random variables; expected value, moments and characteristic functions
Text:
Hwei P. Hsu, *Theory and Problems of Probability, Random Variables, and Random Processes*, Schaum Outline series, McGraw Hill.

CSE 302: Control Systems 3(3,0)

The course familiarizes the student with the mathematical tools and skills needed to analyze and design feedback control systems. Upon completion of this course, students will be able to create mathematical models using block and signal flow diagrams, become familiar with Root Locus techniques, and be able to sketch Root Locus and Bode plots. Design basic feedback systems.
Text: *Modern Control Engineering* by Katsuhiko Ogata, Pearson Education, 4th Edition, 2002.

CS 311: Analysis of Algorithms 3(3,0)

This course covers a number of ideas and techniques useful for designing and analyzing data structures and algorithms. In particular, we will introduce techniques for analyzing upper bounds

for algorithms and lower bounds for problems. Problem areas include sorting, graphs, dynamic programming, combinatorial algorithms, computational geometry, encryption, and NP-Completeness.
Text: *Introduction to Algorithms,* T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. D. Stein, MIT Press.

CS 312: Theory of Automata and Formal Languages 3(3,0)

Theory of computation is the theoretical study of capabilities and limitations of computers. It consists of three branches:

- Automata Theory: Theoretical models of computation
- Computability Theory: What problems are computable (solvable)?
- Complexity Theory: What problems are intractable (highly inefficient to solve)?

The focus of this course is the Automata Theory. However, introductory topics from the other two branches may be included.
Text: *Introduction to the Theory of Computation*, Sipser, 2nd Edition, Thomson Course Technology, 2006.

CSE 321: Electronics-I 4(3,1)

Signals, frequency spectrum of signals, analog and digital signals. Amplifiers, voltage, current and power gains. Gains in decibels, amplifier power supplies, amplifier saturation, non-linear characteristics and biasing. Circuit models for amplifiers. Op-amp configurations and DC imperfections. Diodes, terminal characteristics, modeling of forward characteristics, operation in reverse region and breakdown. Rectifier circuits, limiting and clamping circuits, special diodes. MOSFETs, device structure and physical operation, current voltage characteristics, MOSFET circuits at DC. Biasing in MOSFET and small signal operation and model. BJTs, structure and physical operation, current voltage characteristics. BJT circuits at DC, biasing, small signal operation and models, single stage amplifiers. Differential and multi stage amplifiers.
Text: *Microelectronic Circuits* by Sedra and Smith (latest ed.)

CSE 322: Embedded Systems 4(3,1)

In this course, students will learn fundamentals of embedded system development including both hardware and firmware. The essential issues will be discussed like processor selection, FPGA introduction, hardware/firmware partitioning, circuit design, and circuit layout, coding techniques, circuit debugging, firmware design, firmware architecture, embedded development tools, firmware debugging, real time operating systems and FPGAs.
Text: *Embedded Systems Design: An Introduction to Processes, Tools, and Techniques* by Arnold S. Berger.

CSE 323: Electronics-II 3(3,0)

Feedback, four basic topologies and their analysis. Digital CMOS logic circuits. Memory and advanced digital circuits. Signal generation and wave shaping circuits. Output stages and power amplifiers.
Text: *Microelectronic Circuits* by Sedra and Smith (latest ed.)

CSE 331: Computer Networks 4(3,1)

Computer networks from ISPs to WiFi and cellular networks are a key part of the information economy. These networks are the foundation for the Web, and they enable companies such as Google, Facebook and Amazon. This course introduces the fundamental problems of computer networking, from sending bits over wires to running distributed applications. For each problem, we explore the design strategies that have proven valuable in practice. Topics include error detection and correction, multiple-access, bandwidth allocation, routing, internetworking, reliability, quality of service, naming, content delivery, and security. As we cover these topics, you will learn how the internals of the Internet work to support the Web and other networked applications. You will develop a detailed understanding of widely-used networking technologies such as TCP/IP, HTTP, 802.11, Ethernet, and DNS

Text:

Tanenbaum and Wetherall, *Computer Networks (5th Edition, International version)*

CS 361: Artificial Intelligence 4(3,1)

This is an introductory course in Artificial Intelligence. It includes introduction to knowledge based systems and their structure, usefulness, characteristics, knowledge base, inference engine and user's interface. Knowledge representation methods: semantic nets, frames, rules, logic, depth-first search, breadth-first search and best-first search methods, knowledge elicitation and knowledge acquisition, state spaces and state transition diagrams, essentials of Prolog and an overview to its applications.

Text: *Expert Systems and Applied Artificial Intelligence* by Efraim Turban

CS 362: Operating Systems 4(3,1)

This is designed to be the first undergraduate level course with the main objective of teaching students the concepts and principles that underlie the design and implementation of contemporary operating systems. The discussion of the concepts and principles used in UNIX, Linux, and MS-DOS/MS-Windows operating systems will be integrated throughout the course. Students will be given the opportunity to examine important UNIX and Linux data structures and use its various services through programming exercises. Programming language environment: C on UNIX/Linux.

Text:

1. *Operating System Concepts,* Silberschatz, Galvin and Gagne, 6th Edition, 2002, John Wiley & Sons, Inc.
2. *Modern Operating Systems*, Tanenbaum A.S., 2nd Edition, 2001

CS 363: Database Systems 4(3,1)

The course discusses the fundamental concepts of database design and implementation focusing primarily on relational data model. Structured Query Language (SQL) is also introduced and practiced in the course.

Text: *Fundamentals of Database Systems*, Elmasri & Navathe, 4th Edition.

CSE 371: Signals and Systems 3(3,0)

Signals, signal models, even odd functions. Systems, their classification, models. Laplace transform analysis of continuous time systems. Fourier Series and continuous time system analysis, the Fourier transform. Sampling. Z-transform analysis of discrete time systems. Fourier analysis of discrete time system. Introduction to state space analysis.

Text: *Linear Systems and Signals* by B. P. Lathi (Latest ed.)

CS 381: Software Engineering 4(3,1)

This course is designed to provide understanding of the concepts, techniques and tools for the definition, design, development, testing and maintenance of software systems. The main objective is to emphasize the need for a systematic and managed approach to software development. The course discusses important software development activities, emphasizing on analysis, design, and testing.

Text: *Software Engineering*, Ian Sommerville, 6th Edition, Addison-Wesley, 2001.

CS 382: Web Technologies 3(3,0)

This course will cover many of new and upcoming technology; Student will get hands on practices on different technologies; this course covers Client Side Technologies; HMTL, JavaScript, AJAX and Server Side Technologies; PHP, PHP, Web Services and Rich Internet Application Framework; FLEX Also they will get introduction to upcoming technologies like Semantic Web

Text: This course is a dynamic course being updated each year. Teachers are encouraged to prescribe latest books and material.

CS 401: Computer Graphics 3(3,0)

This course introduces the fundamentals of image processing and computer graphics, which are necessary for the analysis and processing of image data and for the reconstruction, modelling, animation and photo-realistic rendering of 3D. You will learn key computer graphics concepts including graphics primitives, lighting and shading, texture mapping, ray tracing, character animation and curve and surface design.

Texts:

1. Foley, J.D., van Dam, A., Feiner, S.K. & Hughes, J.F. (1990). *Computer graphics: principles and practice*. Addison-Wesley (2nd ed.).

CS 411: Compiler Construction 3(3,0)

The course is intended to teach the students the basic techniques that underlie the practice of Compiler Construction. The course will introduce the theory and tools that can be employed in order to perform syntax-directed translation of a high-level programming language into an executable code. These techniques can also be employed in wider areas of application, whenever we need a syntax-directed analysis of symbolic expressions and languages and their translation into a lower-level description. They have multiple applications for man-machine interaction, including verification and program analysis. In addition to the exposition of techniques for compilation, the course will also discuss various aspects of the run-time environment into which the high-level code is translated. This will provide deeper insights into the more advanced semantics aspects of programming languages, such as recursion, dynamic memory allocation, types and their inferences, object orientation, concurrency and multi-threading.

Text:

Alfred Aho, Monica Lam, Ravi Sethi and Jeffery Ullman, *Compilers: Principles, Techniques and Tools*, Addison Wesley, 2nd edition, 2006.

CSE 421: Computer Architecture 3(3,0)

This course focuses on the principles, practices and issues in computer architecture, while examining computer design trade-offs both qualitatively and quantitatively. Topics include: Advanced Pipelining, Instruction-level parallelism, Memory - hierarchy design, Storage Systems and input/ output and Multiprocessor architecture

Text: *Computer Architecture: A Quantitative Approach*, Hennessy, J. L, and Patterson, D. A., Latest edition.

CSE 422: Digital Design 3(2,1)

The course teaches the design and implementation of digital systems using EDA tools and FPGA. Upon completion of this course, students will have a good comprehension of EDA tool XILINX, become familiar with XILINX FPGA and its programming, programming of Programmable Logic Array and CPLD.

Text: *Advanced Digital Design with the Verilog HDL* by Michael D. Ciletti, Prentice Hall, 2003.

CSE 423: Industrial Automation 3(2,1)

The course offers instruction in electronics applications and computer aspects of automation, including both theory and applications of digital and industrial electronics, hydraulics/pneumatics, actuators, transducers and robotic systems. It also includes discussion of sensors, transducers, signal conditioning devices and computer based data acquisition. Programmable Logic Controllers (PLC) and Ladder Logic are also covered.

Text: *Industrial Control Electronics: Devices, Systems & Applications,* Terry L. M. Bartlet, 3rd Edition, June 2005.

CSE 431: Wireless Networks 3(3,0)

The course deals with the fundamental and practical aspects in the analysis and design of wireless systems. Topics that will be covered are: the wireless communication channel, spectrum efficient digital modulation techniques their comparison and detection techniques, cellular communication principles, techniques used to combat the channel, overview of multiple access techniques and example wireless communication systems.

Text: *Introduction to Wireless Systems* by P. M. Shankar. John Wiley & Sons, 2002.

CSE 432: Internetworking with UNIX TCP/IP

This is designed to be a depth first course with the main objective of teaching students the art of UNIX TCP/IP network programming with BSD sockets using the client-server paradigm. Students gain hands-on experience through the design, implementation, and testing of client-server models for some simple Internet services. Programming language environment: C on UNIX/LINUX.

Text: Douglas E. Comer and David L. Stevens, *Internetworking with TCP/IP: Client-Server Programming and Application, Vol. 3, Linux / POSIX Sockets Version,* Prentice Hall, 2001.

CS 441: Mobile Application Development 3(3,0)

This course is concerned with the development of applications on mobile and wireless computing platforms. Android will be used as a basis for teaching programming techniques and design patterns related to the development of standalone applications and mobile portals to enterprise and m-commerce systems. Emphasis is placed on the processes, tools and frameworks required to develop applications for current and emerging mobile computing devices. Students

will work at all stages of the software development life-cycle from inception through to implementation and testing. In doing so, students will be required to consider the impact of user characteristics, device capabilities, networking infrastructure and deployment environment, in order to develop software capable of meeting the requirements of stakeholders.
Text: To be decided by the instructor.

CS 442: Parallel Programming 3(3,0)

This course explores various aspects of parallel computing including parallel architectures, algorithms, systems, programming languages and implementation issues.
Text: To be decided by the instructor.

CS 445: Programming Languages 3(3,0)

The course is an introduction to the design and implementation of programming languages. From the design point of view, we will study language features as tools for expressing algorithms. From the implementation point of view, we will study compilers, interpreters, and virtual machines as tools to map those features efficiently onto modern computer hardware. The course will touch on a wide variety of languages, both past and present, with an emphasis on modern imperative languages, such as C++ and Java, and, to a lesser extent, on functional languages such as Scheme and ML, and scripting languages such as Perl, Python, and Ruby. Rather than dwell on the features of any particular language, we will focus instead on fundamental concepts, and on the differences between languages, the reasons for those differences, and the implications those differences have for language implementation.
Topics to be covered include:

- formal aspects of syntax and semantics
- naming, scoping, and binding
- scanning, parsing, semantic analysis, and code generation
- control flow, subroutines, exception handling, and concurrency
- type systems, data structures, data abstraction, and storage management
- imperative, functional, logic-based, and object-oriented programming paradigms
- programming environments and tools

Text: Michael L. Scott, *Programming Language Pragmatics*, Elsevier 3rd edition.

CS 461: Management Information Systems 3(3,0)

This course helps you to understand what IT components are available and how you can utilize appropriate IT applications for success. You will learn the terminology used in the field of IT and how IT principles can apply to your businesses. The course stresses the competitive advantage of using IT and the return on investment that you can see. It focuses on the basic principles of Information Technology: hardware and software components, database technology, telecommunications and networking, e-commerce and e-business, Enterprise Resource Planning (ERP), Decision Support Systems (DSS), Artificial Intelligence (AI) and Expert Systems (ES), systems development and implementation, and the ethical and societal issues involved in IT.
Text:
Selected books to be chosen as text book by the course instructor.

CS 462: Data Mining 3(3,0)

Data mining is concerned with the extraction of novel knowledge from large amounts of data. This course introduces and studies the concepts, issues, tasks and techniques of data mining. Topics include data preparation and feature selection, association rules, classification, clustering, evaluation and validation, scalability, spatial and sequence mining, and data mining applications. The course is suitable for undergraduate senior students and graduate students.

Text:

Jiawei Han and Micheline Kamber, *Data Mining: Concepts and Techniques*, Morgan Kauffman Publishers 2000.

CS 463: Bio-informatics 3(3,0)

The objective of this course will be to introduce students to the fundamentals of evolution, molecular biology, and molecular evolution. This course also aims to provide students with a practical and hands-on experience with common bioinformatics tools and databases. It would be expected that after completing this course a student would have a good working knowledge basic bioinformatics tools and databases such as GenBank, BLAST, multiple alignment, and phylogenetic tree construction. Further students will understand the basic theory behind these procedures and be able to critically analyze the results of their analysis using such tools.

Text:

Bryan Bergeron M.D: *Bioinformatics Computing*, 2010.

CS 464: Machine Learning 3(3,0)

This course provides a broad introduction to machine learning, data mining, and statistical pattern recognition. Topics include: (i) Supervised learning (parametric/non-parametric algorithms, support vector machines, kernels, neural networks). (ii) Unsupervised learning (clustering, dimensionality reduction, recommender systems, deep learning). (iii) Best practices in machine learning (bias/variance theory; innovation process in machine learning and AI). The course will also draw from numerous case studies and applications, so that you'll also learn how to apply learning algorithms to building smart robots (perception, control), text understanding (web search, anti-spam), computer vision, medical informatics, audio, database mining, and other areas.

Text: Stanford open course ware.

CS 465: Data Warehousing 3(3,0)

What is Data Warehousing, Inmon vs. Kimball, dimensional modeling, slowly changing dimensions, fact tables, OLAP objects, business intelligence, and extract, transform, and load technologies.

Text:

Ralph Kimball, Margy Ross, Warren Thornthwaite, Joy Mundy and Bob Becker, *The Data Warehouse Lifecycle Toolkit*, Second Edition, 2008. ISBN: 978-0-470-14977-5.

CSE 466: Computer Vision 3(2,1)

Computer Vision aims to extract descriptions of the world from pictures or video. In recent years, much progress has been made on this challenging problem. This course is divided into two parts, Initially first part will introduce basic computer vision concepts. It will then move onto mid-level problems such as tracking and segmentation. Second part will give details about state of the art applications of computer vision such as detecting object classes (e.g. car, table, chair) in images, human action and emotion recognition (walk, run, sad, happy).

Texts:

1. Szeliski, Richard, *Computer Vision: Algorithms and Applications* Springer, 2011.
2. Forsyth, David A., and Ponce, J. *Computer Vision: A Modern Approach*, Prentice Hall, 2003.

CS 467: Ubiquitous Computing 3(3,0)

Ubiquitous computing (ubicomp) is a post-desktop model of human-computer interaction in which information processing has been thoroughly integrated into everyday objects and activities. In the course of ordinary activities, someone "using" ubiquitous computing engages many computational devices and systems simultaneously, and may not necessarily even be aware that they are doing so. This model is usually considered advancement from the desktop paradigm. More formally Ubiquitous computing is defined as "machines that fit the human environment instead of forcing humans to enter theirs.

Text: To be decided by the instructor.

CSE 471: Digital Signal Processing 3(2,1)

The course begins with a discussion of the analysis and representation of discrete-time signals and systems including a discussion of discrete-time convolution, difference equations, the z-transform and the discrete Fourier transform. Considerable emphasis is placed on the similarities with and distinctions between discrete-time and continuous-time signals and systems. The course then proceeds to a consideration of digital network structures for implementation of both recursive (infinite impulse response) and non-recursive (finite impulse response) digital filters. A major consideration in digital signal processing is the design of digital filters to meet prescribed specifications. Thus a set of lectures is devoted to a detailed discussion of digital filter design for both recursive and non-recursive filters. The course concludes with a thorough presentation of the fast Fourier transform algorithm for computation of the discrete Fourier transform.

Texts:

1. *Discrete Time Signal Processing* by Oppenheim, Alan V., and Ronald W. Schafer, 3rd Edition, Prentice Hall, 2010.
2. *Digital Signal Processing: Principles, Algorithms and Applications* by John G. Proakis and Dimitris K. Manolakis, 4th Edition, Prentice Hall, 2006.
3. *Digital Signal Processing: A Practical Approach* by Emmanuel C. Ifeachor and Barrie W. Jervis, Addison-Wesley, 1993.

CSE 472: Communication Systems and Wireless 3(3,0)

The course covers basic principles of communication theory with emphasis on analog modulation systems. Upon completion of this course, students will become familiar with the fundamental features and basic building blocks of analog and digital communications systems, and also be able to characterize the performance of analog modulation techniques in noise

Texts: *Modern Digital and Analog Communication System*s by B. P. Lathi, 3rd edition, Oxford University Press.

Introduction to Wireless Systems by P. M. Shankar. John Wiley & Sons, 2002.

CSE 473: Digital Image Processing 3(3,0)

Concept of digital image, Types of images, Visual Perception, Light & Electromagnetic Perception, Image sensing &acquisition, Spatial and luminance resolution parameters, Image Sampling and quantization, Imaging defects, Noise, Histogram Processing, Spatial Filtering, Convolution & Correlation, Smoothing & Sharpening, Fourier Transform, DFT, Frequency based filtering, Contrast enhancement & adjustment, Noise elimination: smoothing, Histogram manipulation (equalization, compression & Stretching, Image Restoration & Reconstruction, Edge detection, Image segmentation, Segmentation, Feature extraction, Image Coding & Compression, Applications

Text: Gonzalez and Woods, *Digital Image Processing*, Latest edition.

CSE 481: Design Patterns 3(3,0)

This course is an introduction to software design patterns. Each pattern represents a best practice solution to a software problem in some context. The course will cover the rationale and benefits of object-oriented software design patterns. Several example problems will be studied to investigate the development of good design patterns. Specific patterns, such as Observer, State, Adapter, Strategy, Decorator and Abstract Factory will be discussed. Programming projects in the Java language will provide experience in the use of these patterns. In addition, distributed object frameworks, such as RMI, will be studied for their effective use of design patterns.

Text: Gamma, et. Al., *Design Patterns - Elements Of Reusable Object-Oriented Software*, Addison-Wesley, 1995

CS 491 & CSE 491: Final Project 1 3(0,3)

The final project involves research, planning and development of a real and substantial project related to computer science and engineering. It provides an opportunity to the students to use their acquired academic skills in the form of a demonstrable software product. Make oral and written project presentations.

CS 492 & CSE 492: Final Project 1I 3(0,3)

The final project involves research, planning and development of a real and substantial project related to computer science and engineering. It provides an opportunity to the students to use their acquired academic skills in the form of a demonstrable software product. Make oral and written project presentations.