Course Description

Study Plan Academic Cohort entering 2016-2017

(Computer Sciences)

Core Courses

1. **Course number and name:** CS 100T-Discrete Structures

2. **Credits and contact hours:** 3 (3+0+0)

3. **Textbook:**
   a. Main Textbook: " Discrete mathematics and its applications", by K.Rosen Sixth Edition, 2007.
   b. other supplemental materials: " Discrete mathematical structures", by Kolman, Busby & Ross. Sixth Edition, 2008. title, author, and year

4. **Specific course information**
   a. **brief description of the content of the course**
      Brief description of the content: This course covers the basic concepts of discrete mathematics used in computer science and other disciplines that involve formal reasoning. The topics include logic, proof, counting, relations, graphs, trees, and Boolean algebra.

   b. **prerequisites or co-requisites:** None
   c. **Required, elective, or selected elective:** Required

5. **Specific goals for the course**
   a. **specific outcomes of instruction**
      The student will be able to:
   - Introduce the student to the basic mathematical foundation of structures and algorithms that will be used in later computer science courses.
   - Introduce the student to basic concepts in logic, Boolean algebra and proof techniques.
   - Introduce the student to fundamental concepts of computability theory.

   b. **explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.**

      (a) An ability to apply knowledge of computing and mathematics appropriate to the program's student outcomes and to the discipline.

      (b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.

6. **Brief list of topics to be covered**
   - Set Theory
   - Functions, Sequence & Summations
   - Matrices, Processing Arrays
   - Relations
   - Introduction to structured programming: Logic
   - Proof Techniques
   - Recursion

1. **Course number and name:** CS 105T Digital Logic Design

2. **Credits and contact hours:** 3 (3+0+1)

3. **Textbook:**
   a. Main Textbook: "Digital Design", by Mano M. Morris, 4th edition, Prentice- Hall, ISBN 0-13-062121-8, 2007.
   b. other supplemental materials: "Logic And Computer Design Fundamentals", M.Moris Mano and Charles R. Kime, latest edition, 2004.
      – "Fundamentals Of Logic Design", by Charles H. Roth, latest edition, Brooks/Cole Thomson Learning, 2013.
      – "Digital Design, Principles And Practice", by John F.Wakely, Latest Edition, Prentice Hall, Eaglewood Cliffs, NJ, 2005.
      – " Fundamentals Of Digital Logic With VHDL Design", by Stephen Brown and ZvonkoVranesic, McGraw Hill, 2008.

4. **Specific course information**
   a. **brief description of the content of the course**
      Brief description of the content: This course is an introduction to understand the design of digital systems using integrated circuits. The main emphasis is on the theoretical concepts and systematic synthesis techniques that can be applied to the design of practical digital systems. It is intended to give students an overview of the use of Hardware Description Language (HDL) for design basic computer components.

   b. **prerequisites or co-requisites:** None
   c. **Required, elective, or selected elective:** Required

5. **Specific goals for the course**
   a. **specific outcomes of instruction**
      The student will be able to:
      • Understand logic gate operation
      • Demonstrate combinational and sequential circuits
      • Understand the range, precision and errors and their impacts on digital components
      • Convert between different numerical systems.
      • Appreciate the effect of logic gates on binary data.
      • Design a simple circuit using the fundamental building blocks.
      • Synthesize and simulate digital circuits.
   b. **explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.**

      (a) An ability to apply knowledge of computing and mathematics appropriate to the program's student outcomes and to the discipline

      (b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.

      (c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs

6. **Brief list of topics to be covered**
      • Digital Systems and Binary numbers
      • The Digital Systems
      • Binary Numbers
      • Number base conversion
      • Octal and Hexadecimal Numbers
      • Complements
      • Signed Binary Numbers.
      • Signed and unsigned arithmetic
      • Representation of numerical data
      • Binary code.

- Range, precision and error in floating point arithmetic
- Boolean Algebra and Logic Gates
- Basic Definitions
- Axiomatic Definition of Boolean Algebra
- Basic thermos and proprieties of Boolean Algebra
- Boolean Functions
- Canonical and standard forms
- Other Logic Operations.
- Digital Logic Gates.
- Gate Level Minimization
- The Map Method
- Four-Variable Map
- Five-Variable Map
- Sum of Product and Product of sums simplifications
- Don't Care Conditions
- NAND and NOR Implementation
- Other Two-Level Implementation
- Exclusive-OR function
- Combinational Logic
- Combinational Circuits
- Analysis Procedure
- Design Procedure
- Binary Adders-Subtractor
- Decimal Adder
- Binary Multiplier
- Decoders.
- Encoders.
- Multiplexers.
- Use hardware description language HDL
- Synchronous Sequential Logic
- Sequential Circuit
- Storage Elements: Latches
- Storage Elements : Flip-Flops.
- Analysis of Clocked Sequential Circuit.

1. **Course number and name:** CS 110T- Programming language (1)

2. **Credits and contact hours:** 4 (3+2+0)

3. **Textbook:**
    a. Main Textbook: " Java : how to program , P.J. Deitel, H.M. Deitel. -- Latest ed.
    b. other supplemental materials: " Java programming from problem analysis to program design by D.S.Malik -- Latest edition
4. **Specific course information**
    a. **brief description of the content of the course**
        Brief description of the content: This course develops the students' ability to start understanding and writing programs using Java programming language. In this course, students will understand the basic of programming language using java. Decision and control, loops, methods, methods, and create classes will be thoroughly explained and utilized. Dealing with files and streams will be covered as well.
    b. **prerequisites or co-requisites:** None
    c. **Required, elective, or selected elective:** Required

5. **Specific goals for the course**
    a. **specific outcomes of instruction**
        The student will be able to:
        -   Understand Algorithmic problem solving
        -   Recognize and understand the concept of java basic
        -   Understand and implement decision and loops
        -   Recognize and implement methods
        -   Define and implement array
        -   Create and write classes in java

    b. **explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.**

        (a) An ability to apply knowledge of computing and mathematics appropriate to the program's student outcomes and to the discipline.

        (b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.

        (c) Design, implement, & evaluate system, process, component, or program.

6. **Brief list of topics to be covered**
    -   Algorithmic problem solving
    -   java basic
    -   Decision
    -   Loops
    -   Methods
    -   Array
    -   Classe

1. **Course number and name:** CS111T - Programming language(2)

2. **Credits and contact hours:** 4 (3+2+0)

3. **Textbook:**
   a. Main Textbook: " "Java how to program", Paul Deitel, Harvy Deitel.
   b. other supplemental materials: None

4. **Specific course information**
   a. **brief description of the content of the course**
      Brief description of the content: This course develops the students' ability to continue understanding and writing programs using Java programming language. Students will deeply understand the architecture of Object Oriented programming and utilization. Inheritance, Encapsulation, Abstraction, and Polymorphism will be thoroughly explained and utilized. Dealing with files and streams will be covered as well.

   b. **prerequisites or co-requisites:** CS 110T
   c. **Required, elective, or selected elective:** Required

5. **Specific goals for the course**
   a. **specific outcomes of instruction**
      The student will be able to:
      - Student will be able to design, implement, modify, expand, test, and debug java programs written in an object oriented approach.
      - Student will be able to compare and contrast the notions of method's overloading and overriding.
      - Student will be able to develop programs to store user data on different types of files.

   b. **explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.**
      (b) analyze a problem, identify & define computing requirements

      (c) design, implement, & evaluate system, process, component, or program

      (d) function effectively on teams

6. **Brief list of topics to be covered**
   - Classes & Objects
   - Inheritance
   - Polymorphism
   - Exception Handling
   - Files & Steams

1. **Course number and name:** CS 206T-Computer Organization

2. **Credits and contact hours:** 3 (2+2+0)

3. **Textbook:**
   a. Main Textbook: "Computer Organization and Architecture: Designing for Performance,, By William Stallings, latest edition, Prentice-Hall, upper Saddle River.

4. **Specific course information**
   a. **Brief description of the content:**

   This course presents the fundamental principles of Computer Organization and Architecture. Which leads to an understanding of design of processors, the structure and operation of memory and virtual memory, storage system integration, and peripherals. prerequisites or co-requisites
   b. **Prerequisites:** Digital Logic Design - CS 105T
   c. **Required, elective, or selected elective:** Required

5. **Specific goals for the course:**
   a. **Specific outcomes of instruction**

   The student will be able to:

   Explain the significance of current research about a particular topic.

   - Identify the progression of computers from vacuum tubes to VLSI, the concept of an instruction set architecture ISA
   - Understand the difference between microprocessor, microcomputer and microcontroller architectures, and between the various classes of instruction: data movement, arithmetic, logical, and flow control.
   - Recognize the relationship between instruction set architecture, microarchitecture, and system architecture.
   - Understand the way in which subroutines are called and returns made.
   - Define how conditional operations are implemented at the machine level. various types of buses in a computer system and understand how devices compete for a bus and are granted access to the bus
   - Design programs at the assembly language level, to interpret how parameters are passed to subroutines and how local workplace is created and accessed.
   - Demonstrate how interrupts are used to implement I/O control and data transfers
   - Explain the need of open- and closed-loop communications and the use of buffers to control dataflow.

   .

   b. **Explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.**

   (b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.

   (c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs

   (g) An ability to analyze the local and global impact of computing on individuals, organizations, and society

6. **Topics to be covered:**
   - Introduction to Computer System Organization
   - Representation of Instructions
   - Processing Unit
   - Memory Organization and sub system
   - Input/ Output subsystem

1. **Course number and name:** CS 207T-Computer Architecture

2. **Credits and contact hours:** 3 (3+0+1)

3. **Textbook:**
   a. Main Textbook: " Computer organization and Design: The Hardware/ Software Interface" , David A. Patterson and John L. Hennessy, 5th Ed .
   b. other supplemental materials: " Computer Organization and Architecture: Designing for Performance", William Stallings, latest edition, Prentice-Hall

4. **Specific course information**
   a. **brief description of the content of the course**

   Brief description of the content: This course demonstrates the basic metrics by which new and existing computer systems may be evaluated to understand and evaluate the impact of the peripherals, their interconnection and underlying data operations on the design of computer systems. The students will demonstrate the techniques needed to conduct a design of a computer, examine different computer implementation styles and assess their strengths and weakness.

   b. **prerequisites or co-requisites:** Computer Organization CS 206T
   c. **Required, elective, or selected elective:** Required

5. **Specific goals for the course**
   a. **specific outcomes of instruction**
      The student will be able to:
      - Differentiate between the memory technologies found in a computer and be aware of the memory hierarchy and it is necessary to reduce the memory latency.
      - Organize the various cache memories and appreciate its cost-performance and the need for cache coherency in multiprocessor systems
      - Recognize the need for storage standards for complex data storage mechanisms such as magnetic disk, RAIDs, CD, DVD, Blue-ray and HD
      - Explain the register transfer language and the relationship between instruction set architecture, microarchitecture, system architecture and their roles in the development of the computer.
      - Assess the different processor implementation styles and parallel processors taxonomy, assess their strengths and weakness and explain how processor performance can be improved by overlapping the execution of instructions by pipelining and instruction level parallelism ILP.
      - Compute and differentiate between processors performance and system performance by MIPS or SPEC marks and compute how performance can be improved by incorporating multiple processors on a single chip and special-purpose graphics processor GPUs and the relationship between parallelism and performance.
      - Design using hardware description language different units in the processor and execute electronically the computer structures that can be configured and reconfigured.
   b. **explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.**
      (b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.
      (c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs
      (i) An ability to use current techniques, skills, and tools necessary for computing practice.

Brief list of topics to be covered
   - Introduction to the course content, textbook(s), reference(s) and course plan.
   - Introduction; the five components of a computer; performance
   - Introduction to Instruction Set Architecture (ISA) and its design principle
   - Review of register transfer language to describe internal operations in a computer
   - Microarchitectures - hardwired and microprogrammed realizations.

- Different computer implementations
- Instruction pipelining and instruction-level parallelism (ILP)
- Overview of superscalar architectures
- Processor and system performance
- Performance – their measures and their limitations
- Storage systems and their technology
- Storage standards (CD-ROM, DVD)
- Memory hierarchy, latency and throughput
- Cache memories: Caching (Direct mapped cache; Fully associative cache; Set-associative cache; Cache strategies), operating principles, replacement policies, multilevel cache and cache coherency
- Amdahl's law
- Short vector processing (multimedia operations)
- Multicore and multithreaded processors
- Flynn's taxonomy: Multiprocessor structures and architectures
- Programming multiprocessor systems
- Introduction to reconfigurable logic and special-purpose processors.

1. **Course number and name:** CS 212T- Data Structures

2. **Credits and contact hours:** 3 (3+1+0)

3. **Textbook:**
   a. Main Textbook: " Data structures and algorithms in Java", M. Goodrich, R. Tamassia, M. Goldwasser, 2014
   b. other supplemental materials: None

4. **Specific course information**
   a. **brief description of the content of the course**
      Brief description of the content: This course introduce the student to the concept of data structures through abstract data structures including lists, stacks, queues, directed acyclic graphs, and graphs; and implementations including the use of linked lists, arrays, binary search trees, search trees, hash tables, complete trees, and adjacency matrices and lists
   b. **prerequisites or co-requisites:** CS 110T
   c. **Required, elective, or selected elective:** Required

5. **Specific goals for the course**
   a. **specific outcomes of instruction**
      The student will be able to:
      - Apply mathematics in the solution of computing problems
      - Choose appropriate strategy to analyze the problems such as problem decomposition and abstraction
      - Identify the components of algorithmic solutions such as inputs, outputs, variables, types, data
      - Choose an appropriate data structure such as tack, queue, binary tree, or graph required to solve a problem
      - Understand the theory, algorithm and design of algorithmic solutions

   b. **explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.**
      (b) Analyze a problem, identify & define computing requirements

      (c) Design, implement, & evaluate system, process, component, or program

      (k) Design and development principles in the construction of software

         systems of varying complexity.

6. **Brief list of topics to be covered**
   - Arrays
   - Recursion
   - Linked Lists
   - Stacks
   - Queues
   - Trees
   - Graphs
   - Hash Table

1. **Course number and name:** CS 220T Algorithms Design and Analysis

2. **Credits and contact hours:** 3 (3+1+0)

3. **Textbook:**

   a. Main Textbook: "Introduction to Algorithms", by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, Second Edition, 2002.
   b. other supplemental materials:

   Anany Levitin, "Introduction to the Design & Analysis of Algorithms", Addison Wesley, latest "available" Edition.

   Ellis Horowitz, Sartaj Sahni and Sanguthevar Rajasekaran, "Fundamentals of computer algorithms", Galgotia , T2, latest "available" Edition.

4. **Course information:**
   a. **Brief description of the content:**

   This course gives a comprehensive introduction to the design and analysis of algorithms and emphasizes on the techniques for algorithmic problem solving. The focus of this course is on how to design good algorithms, and how to analyze their efficiency. It explains how to compare, and analyze the primary sorting and searching algorithms. It illustrates the behavior of the algorithm with respect to the best, worst, and average case running time. In addition, it describes the brute-force, divide-and-conquer, decrease-and-conquer, transform-and-conquer, greedy, dynamic programming, backtracking, and branch-and-bound methods and their implementations. It covers the essential topics of algorithms design and analysis from a mathematical perspective.

   b. **prerequisites or co-requisites**: CS 212T Data Structures and CS 100T Discrete Structures.

   c. **Required, elective, or selected elective:** Required.

5. **Specific goals for the course:**

   a. **Specific outcomes of instruction**.

   The student will be able to:
   - Recognize the mathematical tools for analyzing algorithms.
   - Describe the required skills (construct, analyze and evaluate) to solve different types of problems correctly and efficiently.
   - Apply the different techniques, skills, and tools that are necessary to solve different problems such as search and sort using asymptotic analysis.
   - Describe and analyze the divide-and-conquer, greedy, and dynamic- programming paradigms.

   b. **Explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.**

   (b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.

   (c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs

   (j) An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices. [CS]

7. **Topics to be covered:**
   - Fundamentals of algorithmic.
   - Problem Solving.
   - Fundamentals of Analysis of Algorithms Efficiency.

- Brute-Force Algorithms.
- Divide-and-Conquer Algorithms.
- Decrease-and-Conquer Algorithms.
- Transform-and-Conquer Algorithms.
- Space and Time Tradeoff in Algorithms.
- Dynamic Programming.
- Limitations of Algorithm Power.

1. **Course number and name:** CS 313T-Advanced Programming language

2. **Credits and contact hours:** 3 (2+2+0)

3. **Textbook:**
   a. Main Textbook: "Visual C# 2012 How to Program", by Paul Deitel, Harvey Deitel, 5th Edition, 2013.
   b. other supplemental materials: None.

4. **Specific course information**
   a. **brief description of the content of the course**
      Brief description of the content: This course deals with the advance concepts and techniques of object-oriented design and programming. C# is the next step after Java. The course examines how to utilize introductory features of the .NET Framework using the C# programming language in order to build basic applications. It includes coverage of .NET and helps the students to develop application programs using C#. Microsoft's .NET architecture is taught in this course as it is needed to understand C#. The course begins by introducing object-oriented programming topics, including inheritance, partial classes and generics. Comprehensive hands-on lab exercises and hands-on projects using Visual Studio enhances lectures and allow easy understanding in the topics being taught.
   b. **prerequisites or co-requisites:** CS 111T
   c. **Required, elective, or selected elective:** Required

5. **Specific goals for the course**
   a. **specific outcomes of instruction**
      The student will be able to:
      - Implement and modify C# programs based on object-oriented programming concepts such as classes, inheritance, polymorphism, overloading, indexers, and exception handling.
      - Develop windows applications using the .NET programming environment.
      - Use the ADO.NET Entity Data Model to establish connections with and manage Transactions of Databases.

   b. **explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.**
      (b) An ability to analyze a problem, and identify and define the computing

         requirements appropriate to its solution.

      (d) Function effectively on teams.

      (k) Design and development of software systems.

6. **Brief list of topics to be covered**
   - Developing Console Applications in the .NET Framework
   - Language Basics
   - Access Modifiers
   - Variables and Data Types.
   - Input/output
   - Expressions and Constants
   - Boxing and Unboxing
   - Implicit and Explicit Casting.
   - Control statements

1. **Course number and name:** CS 340 T-Operating System

2. **Credits and contact hours:** 3 (3+1+0)

3. **Textbook:**
    a. Main Textbook: " Operating System Concepts " by Abraham Silberschatz and Wesley.-Edition, latest Edition.
    b. other supplemental materials: " Operating Systems: A Design -Oriented Approach", by Charles Crowley, The Latest Edition , McGraw-Hill, ISBN: 0256151512.

4. **Specific course information**

    a. **brief description of the content of the course**

    Brief description of the content: This course aims to provide a clear description of the theoretical concepts that underlie operating systems. This course demonstrates the history of the operating systems and provides knowledge of operating systems concepts. The course introduces the principles of processes including inter-process communication, process scheduling, deadlocks, the principles of input / output that includes I/O hardware and software, Memory and File systems management that includes swapping, paging, virtual memory, and page replacement algorithms.

    a. **prerequisites or co-requisites:** Data Structure CS 212 T
    b. **Required, elective, or selected elective:** Required.

5. **Specific goals for the course**

    a. **specific outcomes of instruction**
    The student will be able to:
    - Provide a grand tour of the major operating system components and of basic computer system organization.
    - Describe the services an operating system provides to users, process and other systems and functionality of the operating system software including I/O Drivers.
    - Describe the benefits of a virtual memory system and the physical structure of secondary storage devices.

    b. **explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.**

    (b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.

    (c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs

    (i) An ability to use current techniques, skills, and tools necessary for computing practice.

6. **Brief list of topics to be covered**

    - What Operating Systems Do.?
    - Processes
    - Threads
    - CPU Scheduling Process
    - Deadlocks
    - Memory Management
    - Virtual Memory
    - File-System
    - I/O Systems
    - Mass-Storage Structure

1. **Course number and name:** CS 351T- Human-Computer Interaction

2. **Credits and contact hours:** 3 (3+1+0)

3. **Textbook:**
   a. Main Textbook: "Human computer interaction ", by Dix, J. Finlay, G. Abowd and R. Beale, Third Edition, Prentice Hall, 2014.
   b. other supplemental materials:
   - "Human-Computer Interaction", by Jenny Preece, 1994.
   - "The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies", By Julie A. Jacko, Andrew Sears , Technology & Engineering , 2003.

4. **Specific course information**
   a. **brief description of the content of the course**
      Brief description of the content: This course is an introduction to Human-Computer Interaction concerned with the design, evaluation, and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them. The course considers the inherently multi- and interdisciplinary nature of HCI and situates various HCI issues in the organizational and societal contexts. It introduces theories of human psychology, principles of computer systems and user interfaces designs, a methodology of developing effective HCI for information systems, and issues involved in using technologies for different purposes. It is intended to give students an overview of the entire HCI field by covering most aspects of it.
   b. **prerequisites or co-requisites:** CS 212T
   c. **Required, elective, or selected elective:** Elective or selected elective

5. **Specific goals for the course**
   a. **specific outcomes of instruction**
       The student will be able to:
      - Define the human components functions regarding interaction with computer and Computer components functions regarding interaction with human.
      - Discuss Interaction between the human and computer components.
      - Apply Interaction design basics in information systems development.
      - Evaluate HCI designs of others.
      - Conduct HCI evaluations and usability studies.

   b. **explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.**
      (b) An ability to analyze a problem, and identify and define the computing

      requirements appropriate to its solution.

      (c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs.

6. **Brief list of topics to be covered**
   - The human
   - The computer
   - The interaction
   - Paradigms
   - Interaction design basics
   - HCI in the software process
   - Design rules
   - Implementation support
   - Evaluation techniques.

1. **Course number and name:** CS 360T-Computer Graphics

2. **Credits and contact hours:** 3 (3+0+1)

3. **Textbook:**
   a. Main Textbook : " Interactive Computer Graphics: A Top-Down Approach Using Opengl ", by E. Angel. Addison-Wesley, 2008

      "Computer Graphics With Opengl", by D. Hearn. Pearson Education, 2004.

      "Real-Time Rendering", by T. Akenine-Möller and E. Hines. AK Peters, Latest

      Edition
   b. other supplemental materials: none.

4. **Course information:**
   a. **Brief description of the content of the course**: The course presents fundamental computer graphics techniques. It will help the students understand the concepts and the algorithms of 2D/3D computer graphics and the applications of computer graphics technologies. And it will provide them with a foundation in graphics and Applications programming.
   b. **Prerequisites or co-requisites**: CS 220T Algorithms design and analysis, MASC 242 Principles of linear Algebra.
   c. **Required, elective, or selected elective:** Required

5. **Specific goals for the course:**
   a. **Specific outcomes of instruction**

      The student will be able to:
      - Explain fundamental computer graphics techniques
      - Differentiate between interactive computer graphics techniques and non-interactive techniques
      - Write basic graphics application programs including animation
      - Be capable of using OpenGL
      - Demonstrate and apply concepts and algorithms of 2D/ 3D computer graphics

   b. **Explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.**

      (a) An ability to apply knowledge of computing and mathematics

      (i) use current techniques, skills, and tools

      (k) design and development of software systems

6. **Brief list of topics to be covered**
   - Graphics Models.
   - Graphics Programming
   - Input and Interaction
   - Geometric Objects
   - Geometric Transformations
   - Viewing
   - Shading
   - From Vertices to Fragments
   - Discrete Techniques
   - Programmable Shaders
   - Modeling
   - Curves
   - Surfaces
   - Advanced Rendering

1. **Course number and name :** CS370T Artificial Intelligence

2. **Credits and contact hours:** 3 (3+0+1)

3. **Textbook:**
   a. Main Textbook: Wolfgang Ertel, "Introduction to Artificial Intelligence", Springer, latest Edition
   b. other supplemental materials:Ivan Bratko, "Prolog Programming for Artificial Intelligence", latest Edition.

   Russell and Norvig, " Artificial Intelligence A Modern Approach:, latest Edition

Specific course information

   a. **brief description of the content of the course**
   This course is intended to   acquire basic knowledge and concepts of Artificial Intelligence including knowledge representation, planning and problem solving, logic programming, and learning methods. The focus of this course is on the role of knowledge representation, problem solving approaches, methodologies, and learning in Artificial Intelligence. It describes and characterizes humans' thought processes in Artificial Intelligence.
   b. **Prerequisites or co-requisites:** CS 220D-Algorithms Design and Analysis, CS111D Programming language II

   c. **Required, elective, or selected elective**: Required

4. **Specific goals for the course**
   a. **specific outcomes of instruction**

   The student will be able to:

   * Identify the problems that can be resolved by AI techniques
   * Identify types of Agents
   * Apply problem solving techniques based on constraints
   * Evaluate uninformed and Informed search strategies
   * Differentiate between different forms of learning strategies
   * Explain different knowledge representation, events and objects
   * Apply basic game theory problems such as minMax
   * Design prolog program to evaluate AI problems

   b. **explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.**

   (b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution
   (j) An Ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates Comprehension of the tradeoffs involved in design choices
   (k) An ability to apply design and development principles in the construction of software systems of varying complexity

5. **Brief list of topics to be covered**

   * Intelligent Agents and environments
   * Solving problem by searching
   * Constraint Satisfaction Problems(CSP)
   * Introduction to game theory
   * First order logic
   * Knowledge representation
   * Learning from Examples
   * Introduction to Robotics
   * Prolog syntax and semantics

1. **Course number and name:** CS 380T-Professional Ethics

2. **Credits and contact hours:** 1 (1+0+0)

3. **Textbook:**
   a. Main Textbook: "Ethics for the Information Age", by Michael J.Quinn, Fifth Edition, Pearson Publication.
   b. Other supplemental materials:
      "Localizing the Internet. Ethical Aspects in Intercultural Perspective", by Rafael Capurro, Johannes Fruhbauer, Thomas Hausmanninger (Eds.), Schriftenreihe des ICIE, Band 4, Munich 2007.
      "An Ethical Practice Handbook: A Practical Guide to Dealing With Ethical Issues In Information And Library Work " by David McMenemy and Paul F. Burton (Oxford 2006).

4. **Specific course information**
   a. **brief description of the content of the course**

      Brief description of the content: This course explains the ethical issues that arise as a result of increasing use of computers, and the responsibilities due to work with computers. The course gives a deeper understanding of the social impact of computers and the ethical issues in due to the use of computers. This course also introduces the student to issues like computer crimes and computer security, software theft and intellectual property rights, computer and information system failure, Invasion of privacy and privacy in the Workplace and on the Internet. The course also introduces computer hacking and the creation of viruses.

      This course aims to provide a clear description of the theoretical concepts of the Ethics in daily life. The course demonstrates the history of the essential elements of professional computer society's code of conduct. It also interprets the regional and global consequences of computers and decisions in the societies with ethics. The phrases of this course describe what is expected for the students to know and be able to perform with good knowledge to decide what is right and wrong in general for her life after the completion of an educational program.
   b. **prerequisites or co-requisites:** None
   c. **Required, elective, or selected elective:** None

5. **Specific goals for the course**
   a. **specific outcomes of instruction**
      The student will be able to:
      - Understand the outline on the basic knowledge of computer ethics.
      - Identify common ethical problems in the practice of computing or in computer places.
      - Write the essential code of conduct to her computer place.
      - Interpret the regional and global consequences of computers and decisions in the societies.
      - Provide the necessary background or context Ethics and its importance in life.
      - Suggest an appropriate solution for the any problem in Life.
      - Understand the practical side of ethics related to different theories and differentiate between different characteristics of theories.
      - Focus on the behavior of the learner and not on the learning activity.
      - Can demonstrate essential elements of computer society code of conduct.
      - Predict the possible consequences of the ethical issues in any computer place.

   b. **Explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.**
      i. Understanding of professional, ethical, legal, security and social issues.

6. **Brief list of topics to be covered**
   - Introduction to Ethics
   - Ethical theories – Part 1
   - Ethical theories – Part 2
   - Ethical theories – Part 2
   - Network Communications – Part 1
   - Network Communications – Part 2
   - Intellectual property- Part 1

- Intellectual property- Part 2
- Information privacy-
- Computer and network security- Part 1
- Computer and network security- Part 2
- Computer and network security- Part 3

1. **Course number and name:** CS 385T-Software Engineering

2. **Credits and contact hours:** 3(3+0+ 1)

3. **Textbook:**
   a. Main Textbook: "Software Engineering ", by Ian Somerville, tenth Edition, Addison Wesley, 2015.
   b. Other supplemental materials: "Software Engineering", 7thEdition, McGraw Hill, 2010.

4. **Specific course information**
   a. **brief description of the content of the course**

   This course focuses on the Software Engineering concepts that are needed to develop software systems that can meet basic functional requirements within a well-defined problem domain. It covers the main steps in the process of developing such systems: Requirement Analysis, system specification, system design, and system testing. This course introduces the Object Oriented Analysis and Design (OOAD) approach, and explains the various models of the software that need to be created, the notations to be used for the models (and particularly selected parts of the Unified Modeling Language, or UML). Students participate in teams of Three or four on developing software systems, including the feasibility study requirements, and OOAD.

   b. **Prerequisites or co-requisites:** CS 212T- Data Structure
   c. **Required, elective, or selected elective:** Required

5. **Specific goals for the course**
   a. **specific outcomes of instruction**
      The student will be able to:
      - Apply key elements and common methods for elicitation and analysis to produce a set of Software requirements.
      - Create and specify the software design using a software requirement specification, an Accepted design methodology (e.g., structured or object-oriented), and appropriate design Notation.
      - Use tools necessary for analyses and design activities.
      - Use a software testing strategy.
      - Work in team.
      - Make ethical professional decisions and practice ethical professional behavior.

   b. **Explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course:**
      (b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.
      (d) An ability to function effectively on teams to accomplish a common goal.
      (e) An understanding of professional, ethical, legal, security and social issues and responsibilities.
      (i) An ability to use current techniques, skills, and tools necessary for computing practice.

6. **Brief list of topics to be covered**

   - Introduction to Software and Software Engineering.
   - Software process models.
   - Software Project Management.
   - Software Requirements.
   - Object Oriented Analysis UML Use Case Driven Object Modeling.
   - OO Domain Modeling With UML Class Diagrams and CRC Cards.
   - Behavioral Diagrams: System Sequence Diagrams Collaboration Diagrams.
   - Architectural Design.
   - Software testing.

1. **Course number and name:** CS 430T- Programming Languages Concepts

2. **Credits and contact hours:** 3 (3+0+0)

3. **Textbook:**
    a. Main Textbook: " Concepts of Programming Languages " ,by Robert W. Sebesta, Addison-Wesley, latest edition 2007
    b. other supplemental materials: " Concepts in Programming Languages" , by John C.Mitchell".

4. **Specific course information**
    a. **brief description of the content of the course**
        Brief description of the content: This course develops the students' ability to demonstrate basic knowledge of the fundamental concepts of programming languages and demonstrate basic knowledge of major programming language paradigms imperative and functional and compare major programming language paradigms , imperative and functional, and demonstrate and use syntax, semantics, types and implement subprograms
    b. **prerequisites or co-requisites:** CS 212D
    c. **Required, elective, or selected elective:** Required

5. **Specific goals for the course**
    a. **specific outcomes of instruction**
        The student will be able to:
        *   Describe the importance and power of abstraction in the context of virtual machines.
        *   Demonstrate different forms of binding, visibility, scoping, and lifetime management of variables Practice team work on projects and write technical reports.
        *   Explain how abstraction mechanisms support the creation of reusable and user defined software components.
        *   Describe the different approaches to formal and informal semantics.
        *   Compare and contrast between call-by-value and call-by-reference parameter passing.
        *   Distinguish the main characteristics for each of the programming paradigms covered in this course.

    b. **explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.**

        (b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.

        (e) Understanding of professional, ethical, legal, security and social issues.

        (k) Design and development principles in the construction of software

          systems of varying complexity.

6. **Brief list of topics to be covered**
    *   Describing Syntax and Semantics
    *   Name, Bindings, Type Checking, and Scopes
    *   Data Types
    *   Expressions and Assignment Statements
    *   Statement-Level Control Structures
    *   Subprograms
    *   Survey on programming paradigms.

1. **Course number and name:** CS486T-Web Applications Development

2. **Credits and contact hours:** 3 (2+2+0)

3. **Textbook:**
   a. Main Textbook: " Fundamentals of web development", by Randy Connolly and Ricardo Hoar, 2015.
   b. other supplemental materials: " Web programming step by step", by Marty Stepp & Jessica Miller. Latest edition.

4. **Specific course information**
   a. **brief description of the content of the course**
      Brief description of the content: This course deals with the study of the technologies used to design and develop dynamic web sites. The course will provide students with hands-on experience on professional web-based applications development tools. Topics covered include web page layout with HTML and CSS, JavaScript for event handling, PHP as a server-side scripting language, and MySQL to provide access to a custom database.
   b. **prerequisites or co-requisites:** IS220T
   c. **Required, elective, or selected elective:** Elective.

5. **Specific goals for the course**
   a. **specific outcomes of instruction**
      The student will be able to:
      - Explain the significance of current research about a particular topic:
      - Identify and apply the role, tags and tools used for both Server & Client side web programming to develop simple web pages.
      - Describe the role of CSS in web development and apply CSS for Styling Text, Colors, fonts, and borders of a web page.
      - Describe the scripting constructs of Java script and apply JavaScript to add functionality to web pages and advanced technologies in web development such as AJAX and JQuery.
      - Recognize the main language constructs of PHP to develop dynamic websites.
      - Explain the fundamental principles of Database connectivity to Create database-driven Web applications using MySQL.
      - Work in teams and practice project activities such as writing technical reports.

   b. **explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.**
      (c) An ability to design, implement, and evaluate a computer-based system,

        process, component, or program to meet desired needs.

      (d) An ability to function effectively on teams to accomplish a common goal.

      (i) An ability to use current techniques, skills, and tools necessary for

        computing practice.

6. **Brief list of topics to be covered**
   - Introduction to Web Programming.
   - Web pages and HTML.
   - Web page layout with CSS.
   - Java Script.
   - Server-side programming with PHP.
   - Databases and MySQL.
   - Advanced Topics.

1. **Course number and name:** CS 487T- Graduation Project I

2. **Credits and contact hours:** 2 (2+0+1)

3. **Textbook:**
   a. Main Textbook: "Information Systems Project Management: A Process and Team Approach", by Mark Fuller, Joe Valacich, and Joey George, Sixth Edition, 2017.
   b. other supplemental materials:" Design Patterns: Elements of Reusable Object-Oriented Software", E. Gamma, R. Helm, R. Johnson, and J. Vlissides.

4. **Specific course information**
   a. **brief description of the content of the course**

      The student is required to complete a project that demonstrates his intellectual and comprehensive computer skills and knowledge. This course allows the student to deal with a real and practical computing project. The course enable the student to define and state a problem, figure a solution or solutions to the problem, and then define the system requirements for the problem.

   b. **Prerequisites or co-requisites**: 90 credit hours+ CS 385T- Software Engineering+ IS 350 T– Project Management .

   c. **Required, elective, or selected elective:** required

5. **Specific goals for the course**
   a. **specific outcomes of instruction**

      The student will be able to:

      - Provide the necessary background or context for the project and its importance
      - Propose an appropriate solution for the project problem
      - Provide system design, including the system architecture, implementation requirements and user interface design.

   b. **explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.**

      (b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution

      (c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs.

      (d) An ability to function effectively on teams to accomplish a common goal

      (e) An understanding of professional, ethical, legal, security and social issues and responsibilities

      (f) An ability to communicate effectively with a range of audiences

      (g) An ability to analyze the local and global impact of computing on individuals, organizations, and society

      (h) Recognition of the need for and an ability to engage in continuing professional development

6. **Brief list of topics to be covered**

- Scientific method in research
- Team management
- Time scheduling
- To know what is already known
- Review a system
- Requirement analysis
- System design
- Prototype(building GUI)
- Principles of good writing
- Review some scientific materials

1. **Course number and name:** CS 488T - Graduation Project II

2. **Credits and contact hours:** 2 (2+0+1)

3. **Textbook:**

   a. Main Textbook: depends on the project type, domain, and characterization.
   b. Other supplemental materials: depends on the project type, domain, and characterization

4. **Specific course information**

   a. **brief description of the content of the course**

   The course involves a scientific project in any area of computer Sciences. The course must be taken by a small group of student (4-5). This course is the continuation from the Graduation Project (I).

   b. **Prerequisites or co-requisites:** CS 487T - Graduation Project I
   c. **Required, elective, or selected elective:** Required

5. **Specific goals for the course**

   a. **specific outcomes of instruction**

   The student will be able to:

   - Act responsibly & ethically in carrying out tasks individually as well as group.
   - Work coherently and successfully as a part of a team in assignment.
   - Write technical report and conduct presentation.

   b. **explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.**

   (b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution

   (c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs.

   (d) An ability to function effectively on teams to accomplish a common goal

   (e) An understanding of professional, ethical, legal, security and social issues and responsibilities

   (f) An ability to communicate effectively with a range of audiences

   (g) An ability to analyze the local and global impact of computing on individuals, organizations, and society

   (h) Recognition of the need for and an ability to engage in continuing professional development

   (i) An ability to use current techniques, skills, and tools necessary for computing practice

   (k) An ability to apply design and development principles in the construction of software systems of varying complexity

6. **Brief list of topics to be covered**

- Developing the system based on the design and method proposed in the Graduation Project Proposal.

- Implementing the system.

- Testing the system.

- Designing poster.

- Submission of project documentation and poster.

- Presentation.

1. **Course number and name:** CS 489T- Internship

2. **Credits and contact hours:** 4 (2+0+6)

3. **Textbook:**
   a. Main Textbook: will be assigned depending on the Internship field, and site supervisor's individual student plan.
   b. other supplemental materials: will be assigned depending on the Internship field, and site supervisor's individual student plan.

4. **Specific course information**
   a. **brief description of the content of the course**

   Internship course is an important component of the CCIS - Computer sciences (CS) program. This course is designed to provide an opportunity for CS students to gain a supervised practical experience in computer environment of an approved department, firm or agency in KSA. The students will gain a valuable on-site working experience. It further allows the students to develop skills like communication, teamwork and problem solving skills, which will enable them in joining a competitive job market in their fields. Cooperative Training Office (CTO) should coordinate with students to apply internship. The student and CTO should also submit a written plan for approval before taking-up the internship. All internships are subject to approval by the Internship Coordinator of the college. Note that the maximum number of students is 40 per class.

   b. **Prerequisites or co-requisites:** Completion of 90 Credit hours
   c. **Required, elective, or selected elective:** Required

5. **Specific goals for the course**
   a. **specific outcomes of instruction**

   The student will be able to:
   - Explain current trends and issues specific to the topic of her internship program Practice academic classroom knowledge and skills in a workplace setting.
   - Practice academic classroom knowledge and skills in a workplace setting.
   - Establish a network of acquaintances to increase employment opportunities.
   - Demonstrate an understanding of professional demands (such as behavior, attitude, appearance, and punctuality) of the workplace.
   - Develop appropriate interpersonal and team skills during the course of the internship.
   - Utilize the problem solving skills during the course of her internship.
   - Communicate effectively in verbal and written forms.
   - Demonstrate leadership skills through taking initiatives during internship program.

   b. **Explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.**
   (e) Understanding of professional, ethical, legal, security and social issues.
   (f) Communicate effectively with a range of audiences.
   (h) Continuing professional development.
   (i) Use current techniques, skills, and tools.
   (j) Apply mathematical foundations, algorithms, and theory.

.

6. **Brief list of topics to be covered**

- Introduction to the course content and course overview.
- Internship (every student or group of students will have individual plans that will be made by their respective site supervisor as planned in the internship's environment).

Elective Courses

1. **Course number and name:** CS 321T (Advance Algorithms)

2. **Credits and contact hours:** 3 (3+1+0)

3. **Textbook:**

   a. Main Textbook : '"Introduction to Algorithms", by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein MIT Press and McGraw-Hill, 2017 Edition.
   b. Supplemental materials:" "Network flows:Theory, algorithms, and applications", by Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin, 2017.

4. **Specific Course information:**

   a. **brief description of the content of the course**

   To explain different interaction styles and their use, when to use graphical and textual information presentation and explain principle activities in user interface design process. To introduce usability attributes of system evaluation
   b. **prerequisites or co-requisites**: Algorithms Design and Analysis - CS 220T
   , Data Structures CS 212 T

   c. **Required, elective, or selected elective: selected**

5. **Specific goals for the course:**
   a. **Specific outcomes of instruction**,
   The student will be able to :
   - Develop the association between problem solving, algorithm analysis and complexity analysis.
   - Analyze various cases of problems logically, critically and analytically.
   - Explain and analyze parallel algorithms, primary network flows theory and   NP-hard problems using different techniques

   b. **Explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.**

   (j) An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices
   (k) An ability to apply design and development principles in the construction of software systems of varying complexity

6. **Brief list of topics to be covered:**

   - Fibonacci heaps, Binomial trees, Hashing
   - Maximum and Shortest augmenting path.
   - Ski rental problem, Randomized algorithms.
   - Duality Theory, Linear Programming Algorithms

- Discussing and implementing: Veronoi diagram algorithm.
- Pseudo-polynomial time dynamic programming
- External memory algorithms.Cache oblivious algorithms.

1. **Course number and name:** CS 322T-Formal languages and Automata Theory

2. **Credits and contact hours:** 3 (3+1+0)

3. **Textbook:**
   a. Main Textbook: "Introduction to the theory of computation", by Michael Sipser Second Edition, 2006.
   b. other supplemental materials: Thomas A. Sudkamp, "Languages and Machines", latest "available" edition.

4. **Course information:**
   a. **Brief description of the content:**

   This course covers a comprehensive introduction to automata theory and theory of computation. It discusses Regular sets: finite automata, NFA, DFA, regular expressions, equivalences among notations, and methods of proving a language not to be regular; Context free languages: grammars, pushdown automata, normal forms for grammars, Chomsky NF, and proving languages non-context free; Turing machines: Church-Turing Thesis, equivalent forms, un-decidability; Closure properties, pumping lemmas, and decision algorithms. It describes the basic concepts of complexity classes and the importance of P and NP-completeness.

   b. **prerequisites or co-requisites**: CS 100T Discrete Structures.

   c. **Required, elective, or selected elective:** Selected Elective course in the program.

5. **Specific goals for the course:**
   a. **Specific outcomes of instruction:**
   The student will be able to:

   - Recognize the main concepts of the automata, computability and complexity theory.
   - Describe and Explain CFG, CFL, NFA, DFA, PDA, Chomsky method and pumping lemma for languages.
   - Describe and explain how to classify problems according to their Time - Space Complexity.
   - Apply the efficient techniques, skills, and tools that are required to design and analyze various types of machines such as CFG, CFL, NFA, DFA, PDA, and TM.
   - Describe the properties of various construction methods (transition function and Chomsky formal form and pumping lemma) in automata theory.

   b. **Explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.**

   (b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.

   (j) An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices. [CS]

   (k) An ability to apply design and development principles in the construction of software systems of varying complexity.

6. **Topics to be covered:**
   - Introduction to the course content and course overview.
   - Mathematical notations and basic Terminology.
   - Deterministic, Finite automata (DFA's), and Finite State Machine.
   - Nondeterministic (NFA's - equivalence of DFA's and NFAs).
   - Regular Languages.
   - Regular expressions (regular expression - pumping lemma for regular expressions).

- Context Free Grammars (Context Free Grammar - Parse Trees, Ambiguity in Grammars and Languages, Pushdown Automata - relationship of PDA with CFG - Chomsky Normal Form).
- Turing machine (Church- Turing thesis - nondeterministic TM).
- Variation of Turing Machine.
- Halting problem.
- Decidability, Un-decidability, and Reducibility.
- Classes P and NP (Definition of the classes P and NP.
- Computational complexity theory: Time and Space complexity.

1. **Course number and name:** CS 350T- Multimedia Systems

2. **Credits and contact hours:** 3 (3+0+1)

3. **Textbook:**
   a. Main Textbook: "Multimedia Systems Design", by Andleigh, Prabhat K. & Thakrar K,Prentice Hall, latest edition.
   b. other supplemental materials: "Concepts in Programming Languages", by John C.Mitchell, latest edition.

4. **Specific course information**
   a. **brief description of the content of the course**
   Multimedia systems are an indispensable part of modern information technologies. Students will in the context of this course gain the knowledge and skills necessary for the use and development of multimedia systems. Particular attention will be paid to the software tools for multimedia information (images, audio, and video) representation, coding, storage and transmission, and the compression.
   b. **Prerequisites or co-requisites:** CS 212T-Data Structure.
   c. **Required, elective, or selected elective**: elective

5. **Specific goals for the course**
   a. **specific outcomes of instruction**
   The student will be able to:
   - Recognize the definition of multimedia and characteristics of data streams.
   - Discuss characteristics of analog audio and the psychoacoustic model.
   - Recognize the ability to describe different realizations of multimedia tools.
   - Establish the images, graphics, color, and human visual perception.
   - Design digital video characteristics of motion, persistence of vision.
   - Demonstrate lossless compression information theory.

   b. **explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.**
   (b)Analyze a problem, identify & define computing requirements
   (j)Apply mathematical foundations, algorithms, and theory
   (k)Design and development of software systems.

6. **Brief list of topics to be covered**
   - Introduction to Multimedia Systems.
   - Multimedia Authoring and tools.
   - Audio Technology.
   - Graphics and Images.
   - Color description.
   - Video Technology.
   - Multimedia Compression.
   - MPEG Video Coding.

1. **Course number and name:** CS 361T Computer Vision and Image Processing

2. **Credits and contact hours:** 3 (3+0+1)

3. **Textbook:**
   a. Main Textbook: "Computer Vision: A Modern Approach" , by Forsyth and Ponce, Prentice-Hall, The Latest Edition.

4. **Specific course information**
   a. **brief description of the content of the course**

   Brief description of the content: This course is an introduction to Computer Vision and Image Processing, a discipline concerned with the role of computer vision, signals and their transformation, image processing, image Recognition, images Retrieval and with the study of major phenomena surrounding them. It introduces the principles of Computer Vision and Image Processing, and issues involved in using technologies for different purposes. It is intended to give students an overview of the entire Computer Vision and Image Processing field by covering most aspects of it.

   b. **prerequisites or co-requisites:** Artificial Intelligence CS 370T
   c. **Required, elective, or selected elective:** Selected Elective

5. **Specific goals for the course**
   a. **specific outcomes of instruction**
   The student will be able to:
   - Recall the essential mathematics relevant to computer vision and image processing.
   - Recognize the principles of signal, image processing and object recognition.
   - Define traditional and nontraditional computer vision problems, set goals towards solving them, and observe results.
   - Perform comparisons between (algorithms, methods, techniques...etc).
   - Identify attributes, components, relationships, patterns, main ideas, and errors.
   - Solve computer vision problems using appropriate programming languages and evaluate computer vision systems in terms of general quality attributes and possible tradeoffs presented within the given problem.
   - Recognize the need to engage in continual professional development though the employment of new concepts, techniques, and skills for computing practice.
   - Write technical reports, deliver effective presentations , and communicate effectively with individuals and audiences.

   b. **explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.**
   (b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.
   (f) An ability to communicate effectively with a range of audiences
   (i) An ability to use current techniques, skills, and tools necessary for computing practice.

Brief list of topics to be covered
   - Introduction to Computer Vision and its Applications
   - Fundamentals of Two Dimensional Signals
   - Two Dimensional Transformation
   - Filters
   - Fundamentals of Image Perfection
   - Image Retrieval
   - Image Processing
   - Shapes Recognition
   - Image Center Locating
   - Extracting Image Features
   - Image Features Interpretation & Views Analysis
   - Identification & Tracking of Moving Objects

1. **Course number and name:** CS 372T-Selected Topics I (Data Mining)

2. **Credits and contact hours:** 3 (3+1+0)

3. **Textbook:**
   a. Main Textbook: Jiawei Han, Micheline Kamber, and Jian Pei, Data Mining: Concepts and Techniques, 3rd Edition, Morgan Kaufmann., 2012, ISBN: 978-0-12-381479-1
   b. other supplemental materials: none.

4. **Specific course information**
   a. brief description of the content of the course
   Basic topics of data mining include, data preprocessing, data warehousing and OLAP, data cube, frequent pattern and association rule mining, correlation analysis, classification and prediction, and clustering, as well as advanced topics covering the techniques and applications of data mining in Web, text, big data, social networks, and the design of intelligent decision support system are the detailed topics of study.

   b. Prerequisites or co-requisites: IS 220 T- Database Fundamentals, CS 220 T- Algorithms Analysis and Design, CS 110 T- Programming Language I, CS 111T -Programming Language II
   c. Required, elective, or selected elective: Selected Elective

5. **Specific goals for the course**
   a. specific outcomes of instruction
   The student will be able to:
   - define what data mining is, the components of knowledge discovery process, and data mining applications.
   - Recognize the basic data mining algorithms that support knowledge discovery and decision support systems.
   - Compare alternative data mining algorithms and identify what might be most appropriate for a given data mining task.
   - Use the data preparation tasks and its implications.
   - Demonstrate an understanding of the alternative knowledge representations such as rules, decision trees, association and clustering algorithms.
   - Share ideas and engage effectively on teams to accomplish a common goal either as a member or a leader.

   b. explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.

   (a) An ability to apply knowledge of computing and mathematics appropriate to the program's student outcomes and to the discipline
   (b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.

6. **Brief list of topics to be covered**

   - Course Overview & Introduction
   - Getting to Know Your Data
   - Data Preprocessing - Data Mining Tools
   - Data Warehousing and Online Analytical Processing
   - Mining Frequent Patterns, Associations, and Correlations
   - Classification
   - Cluster Analysis
   - Outlier Detection

- Text Mining
- Big Data

1. <u>**Course number and name:**</u> CS372T - Selected Topics (1) (Data Science and Big Data Analytics)

2. <u>**Credits and contact hours**</u>: 3 (3+1+0)

3. <u>**Textbook:**</u>

   a. Main Textbook: EMC Data Science and Big Data Analytics (Student Guide), Copyright 2013 EMC Corporation.
   b. other supplemental materials: EMC Data Science and Big Data Analytics materials **https://education.emc.com/default_guest.aspx**

4. <u>**Specific course information**</u>
   a. **brief description of the content of the course**
      This course provides an introduction to big data and the Data Analytics Lifecycle to address business challenges that leverage big data. The course provides grounding in basic and advanced analytic methods and an introduction to big data analytics technology and tools, including MapReduce and Hadoop. Labs offer opportunities for students to understand how these methods and tools may be applied to real world business challenges by a practicing data scientist. The course takes an "Open", or technology-neutral approach. The course prepares the student for the Proven™ Professional Data Scientist Associate (EMCDSA) certification exam.
      b. **Prerequisites:** CS212T **and co-requisites**: IS350T, IS 220T, MATH 161
      c. **Required, elective, or selected elective**: Selected Elective

5. <u>**Specific goals for the course**</u>
   a. **specific outcomes of instruction**
      The student will be able to:
      - Explain the importance and benefits of Big Data Analytics and the need for its rapid adoption.
      - Deploy the Data Analytics Lifecycle to address big data analytics projects
      - Use R and RStudio functions
      - Apply appropriate analytic techniques and tools to analyze big data, create statistical models, and identify insights that can lead to actionable results
      - Use tools such as: R and RStudio, MapReduce/Hadoop, in-database analytics, Window and MADlib functions
      - Select appropriate data visualizations to clearly communicate analytic insights to business sponsors and analytic audiences

   b. **explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.**

      (b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.

6. <u>**Brief list of topics to be covered**</u>
   - Introduction to Big Data Analytics
   - Data Analytics Lifecycle
   - Review of Basic Data Analytic Methods Using R
   - Advanced Analytics – Theory And Methods
   - Advanced Analytics - Technologies and Tools

- The Endgame, or Putting it All Together

1. **Course number and name:** CS372T-Selected Topics (1) (Cloud Infrastructure and Services)

2. **Credits and contact hours:** 3 (3+1+0)

3. **Textbook:**

   a. Main Textbook: EMC Cloud Infrastructure and Services (Student Guide - Version 2), Copyright 2014 EMC Corporation.
   b. other supplemental materials: EMC Cloud Infrastructure and Services materials https://education.emc.com/default_guest.aspx

4. **Specific course information**
   a. **brief description of the content of the course**
      This course educates participants on building cloud infrastructure based on a cloud computing reference model. The reference model includes five fundamental layers (physical, virtual, control, orchestration, and service) and three cross-layer functions (business continuity, security, and service management) for building a cloud infrastructure. For each layer and cross-layer function, this course covers the comprising technologies, components, processes, and mechanisms. This course takes an open-approach to describe the concepts and technologies, which are further illustrated and reinforced with EMC-related product examples. The course follows the U.S. National Institute of Standards and Technology as a guide for all definitions of cloud computing.
   b. **Prerequisites or co-requisites:** CS212T and **co-requisites**: CS340T, IS220T , IT221T
   c. **Required, elective, or selected elective:** Selected Elective

5. **Specific goals for the course**
   a. **specific outcomes of instruction**
      The student will be able to:
      - Describe cloud computing, its deployment and service models.
      - Describe the cloud computing reference model and the key considerations to build a cloud infrastructure.
      - Describe the key components and processes required to build physical, virtual, control, and service layers of a cloud infrastructure.
      - Describe the service orchestration, business continuity, security, and service management functions for a cloud infrastructure.

   b. **explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.**

      (b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.

6. **Brief list of topics to be covered**
   - Introduction to Cloud Computing
   - Building the Cloud Infrastructure
   - Physical Layer
   - Virtual Layer
   - Control Layer
   - Service and Orchestration Layers
   - Business Continuity
   - Security
   - Service Management

1. **Course number and name:** CS 372T -Selected Topics (1) (Game Designing)

2. **Credits and contact hours:** 3 (3+1+0)

3. **Textbook:**
    a. Main Textbook: "Game Development Essentials", by Jeannie Novak, Third Edition, 2011.
    b. Other supplemental materials: "Introduction to Game Design, Prototyping, and Development", by Jeremy Gibson, Addison-Wesley Professional, 2014.

4. **Specific course information**
    a. **brief description of the content of the course**

    This course will allow the students an opportunity to study the game development process and theories underpinning the design and development of computer games with a focus on acquiring games programming skills and knowledge.

    b. **Prerequisites or co-requisites:** CS313T -Advanced Programming Language, CS 360 T-Computer Graphics
    c. **Required, elective, or selected elective:** Selected Elective (SE)

5. **Specific goals for the course**
    a. **specific outcomes of instruction**

    The student will be able to:
    - Introduce the students to the history and the factors that pushed computer gaming development.
    - Explain the students the basic mechanics and design structure of digital games
    - Develop a simple computer game in a team environment.

    b. **explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.**

    (a) An ability to apply knowledge of computing and mathematics appropriate to the program's student outcomes and to the discipline

    (k) An ability to apply design and development principles in the construction of software systems of varying complexity.

6. **Topics to be covered:**
    - Introduction and Historical Elements of Computer and Video Games
    - Goals, Genres, Platforms and Player Modes
    - Computer Game Engines (e.g. Unity, Game Maker, etc.)
    - Getting Started with Unity - Terrain, Lights, Skyboxes, Character Controllers
    - 3D graphics for game programming
    - The Game Design principles
    - The Game Design Process and Design Documents
    - Game development
    - Developing 2d and 3d interactive games

1. **Course number and name:** CS 372T-Selected Topics I (Data Mining)

2. **Credits and contact hours:** 3 (3+1+0)

3. **Textbook:**
   a. Main Textbook: Jiawei Han, Micheline Kamber, and Jian Pei, Data Mining: Concepts and Techniques, 3rd Edition, Morgan Kaufmann., 2012, ISBN: 978-0-12-381479-1
   b. other supplemental materials: none.

4. **Specific course information**
   a. **brief description of the content of the course**

   Basic topics of data mining include, data preprocessing, data warehousing and OLAP, data cube, frequent pattern and association rule mining, correlation analysis, classification and prediction, and clustering, as well as advanced topics covering the techniques and applications of data mining in Web, text, big data, social networks, and the design of intelligent decision support system are the detailed topics of study.

   b. **Prerequisites or co-requisites:** IS 220 T- Database Fundamentals, CS 220 T- Algorithms Analysis and Design, CS 110 T- Programming Language I, CS 111T -Programming Language II
   c. **Required, elective, or selected elective**: Selected Elective

5. **Specific goals for the course**
   a. **specific outcomes of instruction**

   The student will be able to:
   - define what data mining is, the components of knowledge discovery process, and data mining applications.
   - Recognize the basic data mining algorithms that support knowledge discovery and decision support systems.
   - Compare alternative data mining algorithms and identify what might be most appropriate for a given data mining task.
   - Use the data preparation tasks and its implications.
   - Demonstrate an understanding of the alternative knowledge representations such as rules, decision trees, association and clustering algorithms.
   - Share ideas and engage effectively on teams to accomplish a common goal either as a member or a leader.

   b. **explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.**

   (a) An ability to apply knowledge of computing and mathematics appropriate to the program's student outcomes and to the discipline

   (b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.

6. **Brief list of topics to be covered**

   - Course Overview & Introduction
   - Getting to Know Your Data
   - Data Preprocessing - Data Mining Tools
   - Data Warehousing and Online Analytical Processing
   - Mining Frequent Patterns, Associations, and Correlations
   - Classification
   - Cluster Analysis
   - Outlier Detection
   - Text Mining

- Big Data

1. **Course number and name** : CS386T-Software Testing

2. **Credits and contact hours** : 3 (2+0+2)

3. **Textbook**
   a. Main Textbook: Introduction to Software Testing , Ammann and Offutt, Cambridge University Press, 2008 or latest edition
   b. other supplemental materials:

4. **Course information:**
   a. **Brief description of the content** :
   This course is intended to develop skills required to test software in structured, organized ways and learn effective, practical ways to design and automate high quality tests during unit and integration testing and learn how to apply theory in practical ways to design tests based on test criteria.

   b. **prerequisites or co-requisites** : Software Engineering – CS385T

   c. **Required, elective, or selected elective:** Selected elective

5. **Specific goals for the course**

   a. **specific outcomes of instruction**:
   The student will be able to:
   - Apply key elements for a test plan for both object oriented software components and web software component
   - Develop software tests according to a test plan
   - Implement a test plan
   - Use tools necessary for reporting results as specified in a test plan
   - Practice team work on projects and write technical reports.

   b. **explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.**

   (b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution
   (i) An ability to use current techniques, skills, and tools necessary for computing practice [CS]

   (d) An ability to function effectively on teams to accomplish a common goal.

6. **Brief list of topics to be covered**

   - Introduction
   - Model-Driven Test Design
   - Test Automation
   - Criteria-Based Test Design
   - Input Space Partitioning
   - Graph Coverage
   - Logic Coverage
   - Managing the Test Process
   - Test Implementation

1. **Course number and name:** CS 387T-Software Design Methodology.

2. **Credits and contact hours:** 3 (2+2+0).

3. **Textbook:**

   a. Main Textbook: Software Engineering, Ian Sommerville, Addison Wesley, latest edition.
   b. Other supplemental materials: E. Gamma, R. Helm, R. Johnson, and J. Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software.

4. **Specific course information:**
   a. **Brief description of the content of the course**
      This course is intended to apply the object-oriented approach and especially component-based software engineering (CBSE) to the design of practical software systems. Construction of large computerized systems proceeds through the stages of requirements analysis and specification, in particular, methodologies for improving the reliability of specifications (executable specifications and system prototypes).
   b. **prerequisites or co-requisites:** Data Structure -CS 212T
   c. **required, elective, or selected elective:** selected elective

5. **Specific goals for the course:**
   a. **Specific outcomes of instruction**

      The student will be able to:

      - Explain the significance of current research about a particular topic:
      - Describe the software process and explain the structured and Object-oriented software development methodologies.
      - Describe the key components in software engineering professional ethics.
      - Elicit, analyze and specify user requirements.
      - Perform object-oriented analysis and formulate the analysis model using modern modeling techniques.
      - Apply object-oriented design principles to make design solutions.
      - Use tools necessary for design activities.
      - Practice teamwork on projects and write technical reports.

   b. **Explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course:**
      (b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.
      (d) An ability to function effectively on teams to accomplish a common goal.
      (e) An understanding of professional, ethical, legal, security and social issues and responsibilities.
      (i) An ability to use current techniques, skills, and tools necessary for computing practice.

6. **Brief list of topics to be covered:**
   - COMPONENT DEFINITION.
   - Visual Modeling with UML.
   - Project Preliminaries.
   - Project Discussion.
   - THE DESIGN OF SOFTWARE COMPONENT INFRASTRUCTURES.
   - SOFTWARE SYSTEMS.
   - Agile Methodology Concepts.
   - THE MANAGEMENT OF COMPONENT-BASED SOFTWARE SYSTEMS.
   - COMPONENT TECHNOLOGIES.

1. **Course number and name:** CS 388T-Agile Software Development

2. **Credits and contact hours:** 3 (2+2+0)

3. **Textbook:**
   a. Main Textbook: "Succeeding with Agile: Software Development Using Scrum", by Mike Cohn, Second Edition, 2010.
   b. Supplemental materials: "A PRACTICAL GUIDE TO THE MOST POPULAR AGILE PROCESS ", by Kenneth S. Rubin, First Edition, 2012.

4. **Specific course information**
   a. **Brief description of the content**:

      This course presents the fundamentals of the process of Agile Software Development. It provides various agile concepts. The focal point of this track is on agile tools, techniques, and methodologies for efficiently and effectively undertaking knowledge-based projects of today "agile methodologies, and managing programs using agile techniques". In addition, it teaches how to properly apply the agile framework in software development process.

   b. **prerequisites or co-requisites**: IS 350T Projects Management.

   c. **Required, elective, or selected elective:** Elective course in the program

5. **Specific goals for the course:**
   a. **Specific outcomes of instruction**:

      The student will be able to :
      - Explain the concepts and process of agile using different tools, principles and techniques.
      - Describe the required skills and techniques to solve different types of tasks appropriately and efficiently.
      - Analyze and apply different types of Agile Software Development process using various tools, methods and techniques.
      - Apply the planning, testing and evaluation in Agile Software Development process*.*

   b. **Explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.**

      (b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution

      (c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs.

      (d) An ability to function effectively on teams to accomplish a common goal.

      (k) An ability to apply design and development principles in the construction of software systems of varying complexity.

6. **Topics to be covered:**
   - An overview of the traditional project management methods.
   - Agile (Meaning, fundamentals and the philosophy of agile. Requirements, architecture, tools and techniques).
   - Project types "projects that are changing all the time".
   - Transitioning to agile and organizational aspects of the agile.
   - Individuals:
     - Overcoming Resistance.
     - New and Changed Roles.
   - SCRUM:
     - Adapting to Scrum.

- Characteristics.
- Framework.
- Concepts.
- Principles.
- Methodology.
- Practice.
- Roles.
- Testing.
- Risk Management.
- Teams:
  - Structure.
  - Teamwork.
  - Self-Organizing team.
  - Sprints.
- Agile Product Management:
  - Communication.
  - Planning.
  - Quality.
  - Measurements.
- The Organization:
  - Scaling.
  - Distributing.
  - Human Resources, Facilities, and the PMO.

1. **Course number and name**: CS390T Modeling and Simulation

2. **Credits and contact hours:** 3 Credits and 3+1 Contact hours

3. **Textbook:** Discrete-Event Simulation: A First Course, Lawrence M. Leemis, Stephen K. Park, 2006.

   Probability and Statistics with Reliability, Queuing, and Computer Science, K.S. Trivedi, 2006.

   a. other supplemental materials:

   1. Performance Evaluation Lecture Notes (Methods, Practice and Theory for the Performance Evaluation of Computer and Communication Systems) Jean-Yves Le Boudec, available online http://ica1www.epfl.ch/perfeval/lectureNotes.htm
   2. System Modelling and Simulation: An Introduction, F. L. Severance
   3. Fundamentals of Queuing Theory, Gross, D. and Carl M. Harris.

4. **Specific course information**
   a. brief description of the content of the course (catalog description): This course enable students to master methodologies for modeling and simulation for complex systems. It also provide understanding of the basic concepts of discrete-event simulation. More over students will be able to analyze the simulation output and validate it. Beside practicing the concept of random number generation and their validation.
   b. prerequisites or co-requisites: Prerequisite MaSc161
   c. indicate whether a required, elective, or selected elective (as per Table 5-1) course in the program: elective

5. **Specific goals for the course**
   a. **The student will be able to:**
      1. Demonstrate methodologies for modelling and simulation of complex systems.
      2. Describe basic concepts of discrete-event simulation.
      3. Analyze the simulation output and validate it.
      4. Demonstrate the concept of random number generation and their validation

   b. **Explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.**
      - An ability to use current techniques, skills, and tools necessary for computing practice (i).
      - An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices (j).
      - An ability to apply design and development principles in the construction of software systems of varying complexity (k).

6. **Topics to be covered:**
   - Introduction to the course content, Textbook(s), reference(s) and course plan.
   - Introduction to simulation and modeling
   - Modelling Methodology
   - Introduction to queues and simulation
   - Statistical models
   - Discrete-event simulation
   - Random Number Generators
   - Mont-Carlo Simulation
   - Randomness Test Goodness of Fit
   - Output Statistics
   - Markov Chains
   - Queuing Theory

- Network of Queues

1. **Course number and name:** CS 408T Parallel Computers Architecture and Programming

2. **Credits and contact hours:** 3 (3+0+1)

3. **Textbook:**

   a. Main text book: John L. Hennessy and David A. Patterson , Computer Architecture: A Quantitative Approach, 5th Edition, Morgan-Kaufmann
   b. other supplemental materials: David E. Culler and Jaswinder Pal Singh, with Anoop Gupta. Parallel Computer Architecture: A Hardware/Software Approach. Morgan Kaufmann, 1998. ISBN: 1558603433.

   Michael J. Quinn. Parallel Programming in C with MPI and OpenMP. McGraw Hill, 2003. ISBN: 0072822562.

4. **Specific course information**
   a. **brief description of the content of the course**

   Brief description of the content: This course appreciates the fundamental principles and trades involved in designing modern parallel computers, as well as the programming techniques to effectively utilize these computers. The course explains the parallel computers, shared data, synchronizing threads, the latency and bandwidth associated with communication. The course exploits the potential processing power of the parallel computers by understanding how to write efficient parallel programs. In addition, It introduces the student to design a good parallel program with understanding how parallel machines are built and vice-versa.

   b. **prerequisites or co-requisites:** Computer Networks Fundamentals IT 221T
   c. **Required, elective, or selected elective:** Elective

5. **Specific goals for the course**
   a. **specific outcomes of instruction**
      The student will be able to:
      - Reorganize the fundamental principles and trades involved in designing modern parallel computers as well as the programming techniques to effectively utilize these computers.
      - Differentiate and contrast architectures for parallel computing, recognizing the strengths and weaknesses of each.
      - Explain parallel computers, shared data, synchronizing threads, the latency and band-width associated with communication.
      - Compare and contrast parallel programming paradigms, recognizing the strengths and weaknesses of each
      - Create, test and debug programs for parallel computing
      - Implement the potential processing power of the parallel computers by understand how to write efficient parallel programs.
   b. **explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.**
      (c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs
      (h) Recognition of the need for an ability to engage in continuing professional development;
      (j) An understanding of processes that support the delivery and management of information systems within a specific application environment.

Brief list of topics to be covered
- Introduction to the course content, textbook(s), reference(s) and course plan.
- Processors in parallel systems
- High performance processor (very large instruction word VLIW processor; vector processors; array processors)
- Caching (Direct mapped cache; Fully associative cache; Set-associative cache; Cache strategies)
- Pipelining
- Supercomputer
- Multicore processors
- Interconnection networks
- Direct connection networks

- Indirect connection networks (busses; Multistage networks; Crossbar switches)
- Distributed memory parallel machines (Multicomputer)
- Virtual shared memory (VSM) and Shared virtual memory (SVM)
- Shared memory parallel machines (Multiprocessors)
- Cache coherency in shared memory machines
- Uniform memory accesses UMA
- Non Uniform memory accesses NUMA
- Cache only memory architecture COMA
- Cache coherency protocols
- Snoopy based multiprocessor
- SIMD array architectures
- Parallel programming overview: (Parallelization process, Challenges, Introduction to Computing, Unified Device Architecture CUDA, MPI; threading; multithreading and Pthreads, Simple programming examples)
- Programming models, mapping computations to parallel hardware, task parallelism, data parallelism, event parallelism, efficient data structures, paradigms for efficient parallel algorithms, and application case studies programming GPU

1. **Course number and name**: CS409T Distributed Systems

2. **Credits and contact hours:** 3 Credits and 3+1 Contact hours

3. **Textbook** Distributed Systems, George Coulouries, Jean Dollimore, Tim Kindberge, 2005.
   a. other supplemental materials1. Distributed Systems: Principles and Paradigms , Andrew S. Tanenbaum, Martin Steen 2002,

4. **Specific course information**
   a. brief description of the content of the course (catalog description): This course introduces students to the complexities involved in designing and building distributed applications. It aims to get learners to gain an in-depth understanding of the principle paradigms used in the area. The course focuses on developing skills of dealing with today's challenging and complexities of distributed systems such as remote invocation, distributed file systems…etc.
   b. prerequisites or co-requisites: Prerequisite CS340T
   c. indicate whether a required, elective, or selected elective (as per Table 5-1) course in the program: elective

5. **Specific goals for the course**
   a. **The student will be able to:**
      1. Describe DS, LANs, WANs, and the evolution of the Internet.
      2. Differentiate between computer networks and distributed systems, and understand challenges w.r.t. distributed systems (scalability challenges with service growing to accommodate new clients).
      3. Introduce Network computing and distributed multimedia, mobile and wireless computing.
      4. Understand the difference between stream and datagrams.
      5. Discuss concepts of the physical layer networking.
      6. Characterize essential protocols, and understand concepts of Data link layer, framing, error control, flow control).
      7. Get familiar with the transport layer services (connection establishment, performance issues).
      8. Write programs that performs any required marshalling and conversion into message units, such as packets.

   b. **Explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course**.

      - An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution (b).
      - An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs ©.
      - An ability to use current techniques, skills, and tools necessary for computing practice (i).

6. **Brief list of topics to be covered:**
- Introduction to the course content, Textbook(s), reference(s) and course plan.
- Characterization of Distributed Systems
- System Models
- Networking and Internetworking
- Inter-process communication
- Distributed Objects and Remote Invocation
- Operating System Support
- Distributed File Systems
- Time and Global States
- Coordination and Agreement
- Fault Tolerant Systems
- Distributed Algorithms and Parallel Computing
- Parallel performance Predication and Measurement
- Programming Languages and Libraries that support parallel and Distributed Systems

1. **Course number and name:** CS410T-Mobile Application Programming

2. **Credits and contact hours:** 3 (2+3+0)

3. **Textbook:**
   a. Main Textbook: "Mobile Applications: Architecture, Design, and Development", Valentino Lee, Heather Schneider, and Robbie Schell, Prentice Hall, 2004.
   b. other supplemental materials: "Mobile Design and Development", Brian Fling, O'Reilly Media, 2009.

4. **Specific course information**
   a. **brief description of the content of the course**
      Brief description of the content: This course covers the topic of how to develop the practical skills needed for coding, tracing, debugging, and building applications that run on Mobile devices both Android and iOS.
   b. **prerequisites or co-requisites:** CS313T
   c. **Required, elective, or selected elective:** Selected elective.

5. **Specific goals for the course**
   a. **specific outcomes of instruction**

      The student will be able to:

      - Explain the significance of current research about a particular topic.
      - Discuss the characterization and architecture of mobile applications.
      - Define mobile computing and the types of mobile devices.
      - Write HTML and JavaScript code for mobile devices.

   b. **explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.**

      (i) An ability to use current techniques, skills, and tools.

      (k) An ability to design and development of software systems.

6. **Brief list of topics to be covered**
   - Introduction to mobile devices
   - Application Development Fundamentals
   - HTML5/JS/CSS3
   - Developing Android Apps
   - Developing iOS Apps
   - Creating projects
   - Configuring the emulator

1. **Course number and name:** CS411T- Advanced Mobile Programming

2. **Credits and contact hours:** 3 (2+3+0)

3. **Textbook:**
   a. Main Textbook: "Programming iOS 5" , by Neuburg, M. Second Edition, 2012.
   b. other supplemental materials: "Learning iPhone programming". Allan, A , Sebastopol, CA : O'Reilly Media, 2010.

4. **Specific course information**
   a. **brief description of the content of the course**
      Brief description of the content: This course covers the topic of how to explain different interaction styles and their use, when to use graphical and textual information presentation and explain principle activities in user interface design process and to introduce usability attributes of system evaluation.
   b. **Prerequisites or co-requisites:** CS410T
   c. **Required, elective, or selected elective:** Selected elective.

5. **Specific goals for the course**
   a. **specific outcomes of instruction**

      The student will be able to:

      - Explain the significance of current research about a particular topic.
      - Recognize advanced programming principles and algorithms to synthesize an innovative mobile solution
      - Critically assess an advanced mobile application in terms of its interface and function.
      - Design, Implement, modify, expand, test, and debug advanced mobile application

   b. **explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.**

      (c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs.

      (i) An ability to use current techniques, skills, and tools.

      (k) An ability to design and development of software systems.

6. **Brief list of topics to be covered**
   - Ubiquitous computing
   - Mobile application development frameworks
   - Design patterns for mobile applications
   - Persistent data in mobile devise environments
   - Utilization of cloud and web services
   - Advanced application deployment
   - Security issues in ubiquitous computing

1. **Course number and name:** CS 412T -System Interface Design

2. **Credits and contact hours:** 3 (2+0+2)

3. **Textbook:**
   a. Main textbook: "The Essential Guide to User Interface Design", by Wilbert O. Galitz—Third edition 2017.
   b. Other supplemental materials:" Link: http://book.protocols.net/

4. **Specific course information:**
   a. **brief description of the content of the course**

      To explain different interaction styles and their use, when to use graphical and textual information presentation and explain principle activities in user interface design process. To introduce usability attributes of system evaluation

   b. **prerequisites or co-requisites:** Human Computer Interaction   CS351T;   Advance Programming CS 313 T
   c. **Required, elective, or selected elective:** Selected elective

5. **Specific goals for the course:**
   a. **Specific outcomes of instruction**

      The student will be able to:
      - Explain the significance of current research about a particular topic.
      - Recognize advanced programming principles and algorithms to synthesize an innovative mobile solution
      - Explain the methods of creation of system interface design based on the user interface elements
      - Designing graphical and web user interface, analyze, test and modify.

   b. **Explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.**

      (b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution

      (c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs.

6. **Brief list of topics to be covered:**

   - Introduction and Overview
   - Design Issues
   - User Interface Design Process
   - Pitfalls in development path, designing for people & Web interfaces
   - Web User Interface, Merging Graphical Business Systems
   - User Interface Elements
   - Test, Retest and Interface Evaluation
   - Scope of Testing, Prototypes, Kinds of Tests

1. **Course number and name:** CS 430T- Programming Languages Concepts

2. **Credits and contact hours:** 3 (3+0+0)

3. **Textbook:**
   a. Main Textbook: " Concepts of Programming Languages " ,by Robert W. Sebesta, Addison-Wesley, latest edition 2007
   b. other supplemental materials: " Concepts in Programming Languages" , by John C.Mitchell".
4. **Specific course information**
   a. **brief description of the content of the course**
      Brief description of the content: This course develops the students' ability to demonstrate basic knowledge of the fundamental concepts of programming languages and demonstrate basic knowledge of major programming language paradigms imperative and functional and compare major programming language paradigms , imperative and functional, and demonstrate and use syntax, semantics, types and implement subprograms
   b. **prerequisites or co-requisites:** CS 212D
   c. **Required, elective, or selected elective:** Required

5. **Specific goals for the course**
   a. **specific outcomes of instruction**
      The student will be able to:
      - Describe the importance and power of abstraction in the context of virtual machines.
      - Demonstrate different forms of binding, visibility, scoping, and lifetime management of variables Practice teamwork on projects and write technical reports.
      - Explain how abstraction mechanisms support the creation of reusable and user defined software components.
      - Describe the different approaches to formal and informal semantics.
      - Compare and contrast between call-by-value and call-by-reference parameter passing.
      - Distinguish the main characteristics for each of the programming paradigms covered in this course.

   b. **explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.**

      (b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.

      (e) Understanding of professional, ethical, legal, security and social issues.

      (k) Design and development principles in the construction of software systems of varying complexity.

6. **Brief list of topics to be covered**
   - Describing Syntax and Semantics
   - Name, Bindings, Type Checking, and Scopes
   - Data Types
   - Expressions and Assignment Statements
   - Statement-Level Control Structures
   - Subprograms
   - Survey on programming paradigms.

1. **Course number and name:** CS 431T-Compiler Design

2. **Credits and contact hours:** 3 (3+0+0)

3. **Textbook:**
   a. Main Textbook: " Compilers: Principles, Techniques, and Tools",   by Alfred V. Abo, Monica S. Lam, Ravi Sethi, and Jeffrey D. Ullman, ISBN-13: 978-0321486813, ISBN-10: 0321486811, 2014, Widely known as the "Dragon book". Pearson Education.
   b. other supplemental materials:
      i. "The Compiler Design Handbook: Optimizations and Machine Code Generation", by Y.N. Srikant and Priti Shankar, Second Edition, 2007.
      ii. "Modern Compiler Design",by David Galles, 2004.
      iii. http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-035-computer-language-engineering-spring-2010/.

4. **Specific course information**
   a. **brief description of the content of the course**
      Brief description of the content: This course introduces the student to the fundamental concepts, functions, and structures of compiler construction. It describes the phases of program translation from source code to executable code and analyses the most important techniques for the representation and generation of Languages.
   b. **prerequisites or co-requisites:** CS340T
   c. **Required, elective, or selected elective:** Elective

5. **Specific goals for the course**
   a. **specific outcomes of instruction**
      The student will be able to:
      - Illustrate the components, notations concepts, terminology and architecture of a compiler.
      - Describe the details of the different phases of the compilation process: lexical analysis, grammars and parsing, intermediate representations, syntax-directed translation, code generation, type checking, simple Dataflow and control-flow analyses, and optimizations.
      - Compare and contrast between compiled versus interpreted programming languages.
      - Use regular expressions and grammars, finite automata and context-free grammars in syntax diagrams.
      - Apply different techniques and algorithms for optimizing executable code.

   b. **explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.**
      (a) An ability to apply knowledge of computing and mathematics appropriate to the program's student outcomes and to the discipline.
      (b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.
      (h) Recognition of the need for and an ability to engage in continuing professional development.

6. **Brief list of topics to be covered**
   - Introduction to the Notion of Compiler
   - Regular expressions and language specification by formal grammars.
   - Top-Down Parsing: Left Recursion and Left Factoring, First Sets, Follow Sets, Recursive Descent Parsing, and LL(1) Table-Driven Parsing
   - Bottom-Up Parsing: SLR, LAR and LR Parser
   - Parser Generator: Lex and Yacc (or Jlex and Cup)
   - Semantic Processing: Syntax-Directed Translation, Intermediate Representations, and design of Predictive Compilers.
   - Error Detection and Recovery.
   - Intermediate Code Generation. Foundations of data-flow analysis.
   - Code optimization. Instructions of optimization. Un-optimized code generation.
   - Introduction to Register allocation and Parallelization.

1. **Course number and name:** CS 460T-Principles of Computer Animation

2. **Credits and contact hours:** 3 (2+0+2)

3. **Textbook:**
   a. Main Textbook: R. Parent, Computer Animation, latest Edition: Algorithms and Techniques, Morgan Kaufmann.
   b. other supplemental materials: Advanced Animation and Rendering Techniques by Watt and Watt, Addison Wesley.

4. **Specific course information**
   a. **brief description of the content of the course**

   Brief description of the content: This course demonstrates how to understand and implement computer animation algorithms used to specify and generate motion for graphical objects which play significant role in computer simulation and video game software..
   b. **prerequisites or co-requisites:** Computer Graphics CS 360T
   c. **Required, elective, or selected elective:** Elective

5. **Specific goals for the course**
   a. **specific outcomes of instruction**
      The student will be able to:
      - Express knowledge of fundamental techniques used for computer animation.
      - Express knowledge of fundamental techniques used for computer animation.
      - Express knowledge of fundamental techniques used for computer animation.
      - Express knowledge of fundamental techniques used for computer animation.
      - Identify the advantages and disadvantages of using simulation, procedural animation, motion capture, and hand designed animations, with respect to both online and offline applications.
      - Identify the advantages and disadvantages of using simulation, procedural animation, motion capture, and hand designed animations, with respect to both online and offline applications.
      - Show the ability to function effectively on teams to develop computer animation tools.
   b. **explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.**
      (c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs
      (k) An ability to apply design and development principles in the construction of software systems of varying complexity.

6. **Brief list of topics to be covered**
   - Introduction
   - Motion Perception
   - The Heritage of Animation
   - Animation Production
   - Computer Animation Production
   - A Brief History of Computer Animation Technical Background
   - Spaces and Transformations
   - Orientation Representation
   - Interpolating Values
   - Interpolation
   - Controlling the Motion of a Point along a Curve
   - Interpolation of Orientations
   - Working with Paths
   - Interpolation-Based Animation
   - Key-Frame Systems
   - Animation Languages
   - Deforming Objects
   - Three-Dimensional Shape Interpolation
   - Morphing (Two-Dimensional)
   - Kinematic Linkages
   - Hierarchical Modeling
   - Forward Kinematics

- Inverse Kinematics
- Motion Capture
- Physically Based Animation
- Fluids—Liquids and Gases
- Modeling and Animating Human Figures
- Facial Animation
- Behavioral Animation Overview
- Special Models for Animation

1. **Course number and name: :** CS 461-T Computational   Geometry

2. **Credits and contact hours:** 3 (0+3+0)

3. **Textbook:**
   a.   Main Textbook: " Mark de Berg · Otfried Cheong Marc van Kreveld · Mark Overmars
   b.   other supplemental materials: Computational Geometry Algorithms and Applications Third Edition

4. **Specific course information**
   a. **brief description of the content of the course**
      Brief description of the content: This course develops the students' ability to demonstrate basic knowledge of the fundamental Computational Geometry and demonstrate basic knowledge of major Line Segment Intersection Thematic Map Overlay and Orthogonal Range Searching Querying a Database and Statement-Level Control Structures, and Convex Hulls Mixing Things

   b. **prerequisites or co-requisites:** Data Structure CS 212 T
   c. **Required, elective, or selected elective:** Elective

5. **Specific goals for the course**
   a. **specific outcomes of instruction**
      The student will be able to:
      - Understand the fundamental problems within computation geometry and general
      - Techniques for solving problems within computational geometry
      - Practical experience with implementation issues involved in converting computation

   b. **explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.**

      (b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution

      (c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs

         .

6. **Brief list of topics to be covered**
   - Computational Geometry Introduction
   - Line Segment Intersection Thematic Map Overlay
   - Linear Programming Manufacturing with Molds
   - Orthogonal Range Searching Querying a Database
   - Statement-Level Control Structures
   - Convex Hulls Mixing Things
   - Convex Hulls Mixing Things
   - Matrices, Processing Arrays
   - Relations

1. **Course number and name:** CS 471T-Robotics Fundamentals

2. **Credits and contact hours:** 3(3+1+0)

3. **Textbook:**

   a. Main Textbook: "Introduction To Ai Robotics",by R. Murphy. A Bradford Book, MIT Press. Latest Edition.
   b. Supplemental materials: "Behavior-Based Robotics", by Ronald C. Arkin. MIT Press. Latest Edition
      "Understanding Intelligence", by R. Pfeifer and C. Scheier.. MIT Press. Latest Edition

4. **Specific Course information:**

   a. **Brief description of the content:**
      The general notion of situated agent is presented by introducing behavior-based systems, the direct coupling of perception to action and the dynamic interaction with the environment. A number of examples from industrial robotics and autonomous vehicle control are discussed with some experimental lectures using available moving platforms. The locomotion problem for the so called legged-robots is also presented discussing various kinds of mechanical structures, humanoids included.

   b. **prerequisites or co-requisites**: Artificial Intelligence CS 370T
   c. **Required, elective, or selected elective:** Required

5. **Specific goals for the course**

   a. **specific outcomes of instruction**,

      The student will be able to:
      - Outline the potential and limitations of today's state-of-the-art robot systems
      - Describe various strategies for navigation in unknown environments, including the strengths and shortcomings of each.
      - Implement configuration space algorithms for a 2D robot and complex polygons
      - Explain how an agent differs from other categories of intelligent systems.
      - Justify programs which are able to control autonomous vehicles.
      - Judge and manipulate the most important techniques related to robotics.
      - Evaluate the basic knowledge of behavior-based systems, the direct coupling of perception to action and the dynamic interaction with the environment.
      - Demonstrate the locomotion problem for the so called legged-robots.

   b. **explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.**

      (b) analyze a problem, identify & define computing requirements

      (c) design, implement, & evaluate system, process, component, or program

      (i) use current techniques, skills, and tools

6. **Topics to be covered:**

   - Introduction to the course content, Textbook(s), reference(s) and course plan.
   - Situated Agents
   - Sensing and Acting
   - Autonomous Vehicles, Mobile Robotics, Control Issues
   - Braitenberg's Vehicles, Applications

- Dynamical Agents, Dynamical System Review, Mobile Vehicles
- Representational Issues, Energy-based Methods for Movement Controls
- Implementing Behaviors, Perceptual Basis for Behavior-based Control, Representational Issues
- Reactive Architectures, Hybrid Architectures, Adaptive Behaviors, Collective Behaviors
- Locomotion, Wheel-driven vehicles, Legged robots, four-legged and bipedal robots
- Humanoid: a brief introduction, Active and Passive Walking, Gait control using the inverted pendulum model
- Experimental Activity, Using LEGO Mindstorm