6001101-3 - Introduction to Computer Science

Credits and contact hours: 3 Credits (2 x 50 mins lectures, 2 x 50 mins labs)

Instructor:

Textbook:

- *"Computer and Software Packages", Dr. Mohammed Bilal Al Zoubi, et al*
- *"How to Program,"Dietel, M., Prentice Hall*

***Specific courseinformation*:**

a. *Enable* the student to understand the basics of Computer Science and the understanding of the principles of programming and the use of a computer to solve some scientific issues.
b. Prerequisite None
c. *Required,* elective*, or selected elective:*

***Specific goals for thecourse*:**

a. Specific outcomes of instruction:

1. *Students learn basics of computers hardware and software*
2. *Students learn a few of the internal workings of computers and numbering systems*
3. *Learn how to solve basic programming problems using simple concepts*

(A): An ability to apply knowledge of mathematics, computing, science, and engineering appropriate to the discipline;

Students apply knowledge of computing and design to a project.

Topics covered:

- A general idea about the components of computer and functions of each of them and its components, characteristics of computers, classification of computers and its advantages.
- Numbering systems and conversions between systems and numerical operations on them.
- The representation of data within a computer memory
- Fundamentals of programming and problem solving techniques and algorithms and flow charts of operations

A detailed introduction to C/C++ (basic fundamentals, I/O operations, arithmetic operations and conditional statements)

6001103-3- Computer Programming

Credits and contact hours: 3 Credits (2 x 50 mins lectures, 2 x 50 mins labs)
Instructor: Ahmed Alaloi

Textbook:

- **Head First Java, 2nd Edition by Kathy Sierra and Bert Bates.**

***Specific courseinformation*:**

a. *Introduce students to the basics of writing software programs including variables, types, arrays, procedures, control structures, input/output, and general rules for writing good code.*
b. *Prerequisite*: *6001101-3* - Introduction to Computer Science

***Specific goals for thecourse*:**

a. *Specific outcomes ofinstruction*:
    a. Students will see how programming languages have developed, appreciating their continuous development, and realise the need for life-long learning.
    b. They will be able to write and debug small programs (50-100 lines) using a range of procedural techniques using a variety of input sources (console, file) and a variety of output destinations (console, file).
    c. Students will learn basic design principles that are relevant to this style of "programming in the small" including decomposition, elimination of redundancy, detailed documentation, and use of parameters and return values to create flexible components.
b. *ABET Criterion 3 Student Outcomes addressed by thecourse*:

(A): An ability to apply knowledge of mathematics, computing, science, and engineering appropriate to the discipline;

Students apply knowledge of computing and design to a project .

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

Students learn the basics of gathering requirements for solving a problem

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

Students are required design and implement a software project to meet a specification .

(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.;

Students are required to apply their knowledge of computing to design a small procedural program.

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

The students are required to use standard design and development principles on a significant software project .

Topics covered:

- Basic concepts of writing code, compilation, and execution
- Variables, types, assignment, and expressions
- Defining methods: parameters, return values
- Conditional execution (if/else, case)
- Iteration (for, while)
- Arrays (one-dimensional)
- Input/output: console, file
- Procedural decomposition of problems

Programming style: eliminating redundancy, localizing variables, constants, commenting, use of parameters and return values to increase flexibility.

6001104-3 - Structured Programming

Credits and contact hours: 3 Credits (2 x 50 mins lectures, 2 x 50 mins labs)

Instructor:

Textbook:

- **"Head First Java, 2nd Edition by Kathy Sierra and Bert Bates**

***Specific courseinformation*:**

a. Students will experience in advanced programming using the Java language, which The course aims to provide students with advanced skills in programming depending on what he had learned the basics of programming, to be able to link the programs of health data entered to the system and word processing and identification of errors and exceptional treatment and the definition of new varieties to deal with exceptional errors. Students will also advance their programming skills by exploring good design principles (e.g., patterns), understanding bad design methods (e.g., anti-patterns), and gain early experience in O.O.P such as design, specification, and implementation

b. *Prerequisite*: 6001103-3- Computer Programming

c. Required, elective, or selected elective:

***Specific goals for thecourse*:**

a. *Specific outcomes ofinstruction*:

1. Students will be able to apply standard data abstractions (lists, maps, sets, stacks, queues) and recursion to solve straightforward programming problems
2. Students will appreciate some of the advantages of, and be able to apply, object oriented design (encapsulation, interfaces, inheritance)
3. Students will appreciate the benefits of code reuse by learning how to make use of off-the-shelf components from libraries such as the Java Collections Framework.

b. *ABET Criterion 3 Student Outcomes addressed by thecourse*:

(A): An ability to apply knowledge of mathematics, computing, science, and engineering appropriate to the discipline;
Students apply knowledge of computing and design to a project .

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

Students learn the basics of gathering requirements for solving a problem

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

Students are required design and implement a software project to meet a specification .

(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.;

Students are required to apply their knowledge of computing to design a small procedural program.

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

The students are required to use standard design and development principles on a significant software project .

Topics covered:

- Implementing linked structures (linked lists, binary trees)
- Abstract data types: stacks, queues, lists, maps, sets
- Recursion and recursive backtracking
- Using off-the-shelf components (e.g., Java Collections Framework)
- Use of inheritance for additive change and factoring out common code
- Exceptions and exception handling
- Class design: encapsulation, documentation, appropriate choice of fields
- Thorough testing and debugging

Virtual machines and intermediate languages

6001210-3 - Discrete Structures

Credits and contact hours: 3 Credits (2 30-minutes lectures per week)

Instructor: **Amin Daoud**

Textbook: KR] Discrete Mathematics and Its Applications, 4th Edition, By Kenneth Rosen
[JMJN] Invitation to Discrete Mathematics, 2nd Edition, By Jiri Matousek and JaroslavNesetril

***Specific course information*:**

a. This course covers the mathematical foundations of computer science and engineering. It provides an introduction to elementary concepts in mathematics such as definitions, logic, proofs, functions, relations and countingprinciples.Thecoursealsointroducesstudentstoelementarydiscrete structures such as sets, partial orders, graphs andtrees

b. *Prerequisite*: 404151-4 – Introduction to SetTheory
c. *Required, elective, or selectedelective:*

***Specific goals for the course*:**
Specific outcomes of instruction:

a. Be able to analyze complexity ofalgorithms
b. Be able to apply number theory to practicalproblems
c. Be able to synthesize elementaryproofs

d. *ABET Criterion 3 Student Outcomes addressed by thecourse*:

(A): an ability to apply knowledge of mathematics, computing, science, and engineering appropriate to the discipline;
Students will be able to analyze computational processes

(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.;
This course provides the foundation for algorithms and theory of computing. Hence the students will be able to apply methods learned in this course to analyze and reason mathematically about the tradeoffs involved in design choices. Furthermore this course will enable students to model many systems using discrete structures.

Topics covered:

- Logic, Truth Table, Propositionalequivalences
- Predicates andQuantifiers
- Sets andFunctions

- Relations, Equivalences and PartialOrders
- Proofs: Induction, Contradiction, Contrapositives
- Counting Principles: Cardinality, factorials, permutations, Binomial coefficients, Inclusion-Exclusion, Pigeon-Hole Principle, sums and a asymptotic
- Graphs and Trees: Representation, degree sequences and hand shakinglemma, Euler tours, Planar graphs, Euler Formula. Properties of Tree, SpanningTrees

6001222-3: System Analysis & Design

Credits and contact hours: 3 Credits (2 30-minutes lectures perweek)

Instructor: Khalil Alsulbi

Textbook :
Systems Analysis and Design Methods, 6th Edition by Jeffery L. Whitten, Lonnie D. Bentley and Kevin C. Dittman, 2004, McGraw-Hill

***Specific courseinformation*:**

a. To introduce students to the relative complexity of information requirements, systems analysis and design within a business organization, and to introduce students to the concepts, formal techniques, tools and methods used in the analysis, design and implementation of information systems. The course approaches the development of information systems from a problem-solving perspective. This course builds upon concepts to which the student has been exposed to in previous classes.
b. *Prerequisite*: 6001101-3 - Introduction to Information Systems.
c. Required, elective, or selected elective:

***Specific goals for thecourse*:**

a. *Specific outcomes ofinstruction*:
    a. Know the notion of a system and the phases, activities and deliverables in system analysis
    b. Know the basic techniques of systems analysis, design and implementation
    c. Understand and synthesize the different models used to describe a system, the competencies needed by systems analysts to carry out their tasks and responsibilities successfully, and fact finding and analysis techniques used in system analysis
    d. Be able to perform system analysis, and work successfully with team members *ABET ABET.*
b. *Criterion 3 Student Outcomes addressed by thecourse*:

(A): an ability to apply knowledge of mathematics, computing, science, and engineering appropriate to the discipline

Students apply knowledge of computing and design to a project

(B): an ability to analyze a problem, and identify and define the computing requirements appropriate to its solution

Students apply systems analysis to gather system requirements

(D): an ability to function effectively on teams to accomplish a common goal

Projects are implemented in teams.

(F): an ability to communicate effectively

The projects require communications, specifications, progress reports, and final report

(H): a recognition of the need for, and an ability to engage continuing professional development

The students often must utilize the internet to learn and apply the new technologies that they have chosen in support of their projects.

(K): an ability to apply design and development principles in the construction of software systems of varying complexity.

Students use standard design and development principles on a significant software project.

Topics covered:

- Introduction and Overview
- Software development process, object-oriented development
- The requirements engineering process: Facilitated workshops, Fact-finding interviewing, Other requirements elicitation techniques
- Documenting requirements, analyzing requirements
- Requirements Management, Validating requirements
- Feasibility Analysis and System Proposal
- Systems Design, Data Flow Diagrams, Process Modeling, Database Design
- Output Design And Prototyping
- Input Design And Prototyping
- System Constructions and Implementation

6001227-3 - File Processing and organization

Credits and contact hours: 3 Credits (2 30-minutes lectures per week)

Instructor: HamzaAwadElkarimHamza

Textbook: Introduction to Algorithms by T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein, Second Edition, MIT Press, 2001. ISBN0-262-03293-7

***Specific course information*:**

a. Design and analysis of efficient computer algorithms. Algorithm design techniques, including divide-and-conquer, depth-first search, and greedy approaches. Worst-case and average-case analysis. Models of computation. NP-completeproblems..

b. *Prerequisite*: 6001218-4 - Algorithms and DataStructures

c. *Required, elective, or selectedelective:*

***Specific goals for the course*:**

a. The students will be able to describe, construct, and use various implementations for advanced data abstractions such as morespecialized search trees andheaps.
b. The students will be able to design and implement advancedalgorithms and analyzethem.
c. The students will develop an understanding of variousalgorithm-design paradigms e.g. divide-and-conquer, greedy,etc.

b. *ABET Criterion 3 Student Outcomes addressed by thecourse*:

(A): An ability to apply knowledge of mathematics, computing, science, and engineering appropriate to the discipline;

Students will develop an understanding of how to represent different algorithm resource requirements as mathematical functions on the size of the input (logarithmic, linear, etc.)

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

Students can analyze the time and space requirements of a particular problem by performing asymptotic analysis.

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

Studentsareaskedtomodifyalgorithmstoproducedifferentoutputsorcombinealgorithms and data structures to offer newsolutions.

(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.

Students are able to compare various algorithms for specific problems and the optimal choice of data structures.

Topics to be cover:

- AdvancedSearch-TreeStructures(Red-BlackTrees,B-Trees,Tries,SplayTrees). Arithmetic expression processing using astack.
- Advanced Heap Structures (FibonacciHeaps).
- Graphs and Graph Algorithms (Graph Representations, Depth-First Search, Breadth-First Search, Minimum Spanning Trees, Shortest Paths, Maximum Flow, Matching).
- Geometric Algorithms (Intersection of Line Segments, ConvexHull).
- Advanced Design and Analysis Techniques (Greedy Algorithms, Dynamic Programming)

6001217-3 - Logic Programming

Credits and contact hours: 3 Credits (2h 30-minutes lectures perweek)

Instructor: Mohamed Amin HADJ TAIEB

Textbook:

- [Programming in Prolog: Using the ISO Standard Clocksin and Mellish, Springer, 2003, ISBN 3540006788

***Specific courseinformation*:**

c. The aim of this course is to present the key concepts behind logic programming: logic as a declarative (context-free) language, how to write programs with logic, and how to make efficient implementations. In particular, we will cover: recursive structures, syntax and semantics of propositional logic, 1st order and higher-order logics, inferences rules, unification and resolution, SLD-resolution, negation as failure, and implementation issues.

d. Prerequisite:
6001210-3 – Discrete Structures

e. Required, elective, or selected elective:

***Specific goals for thecourse*:**

c. *Specific outcomes ofinstruction*:
   1. Students will appreciate the declarative programming model, and be able to identify when it would be useful in problem solving.

d. *ABET Criterion 2 Student Outcomes addressed by thecourse*:

(A): An ability to apply knowledge of mathematics, computing, science, and engineering appropriate to the discipline;
Students apply knowledge of computing and design to a project.

(I):*An ability to use the current techniques, skills, and tools necessary for computing practice;*

Projects use current computing and modeling/design tools.

Topics covered:

- Introducing the concepts of Logic, Mathematical Logic, and Logic programming.
- Distinguishing between declarative, object-oriented, and logic programming methodologies.
- Directionless of logic programming.
- Mathematical Logic (First order logic) issues (Rules of mathematical representation, Representing facts and rules)

- Mathematical Logic (First order logic) issues (Deduction, Computation function and predicate, Unification, Resolution, and Clause Form)
- Introduction to Logic Programming and Prolog Syntax.
- Starting Prolog with Prolog terms and Prolog programs.
- Clauses, Predicates, Variables.
- Common Variables and Satisfying and evaluating goals.
- Unification and backtracking.
- Operations and Arithmetic. Input and Output.
- Loops, Preventing Backtracking.
- Lists and String in Prolog.

6001211-3 - Web Programming

Credits and contact hours: 3 Credits (2h 30-minutes lectures per week)

Instructor: Mohamed Amin HADJ TAIEB

Textbook:

- Primary Book: Java Network Programming, Elliotte Rusty Harold, O'Reilly,3rd edition, 2005, ISBN:978-0-596-00721-8
- Secondary Book: Jan Graba, An Introduction to Network Programming withJava, 2nd edition, 2007, ISBN-13:978-1-84628-380-2

***Specific course information*:**

a. This course provides an introduction to network/Internet programming. It covers the major concepts for programming distributed applications, in particular, asynchronous and synchronous inter-process communication, process synchronization and remote procedure call(RPC).

b. Prerequisite: 6001102-3 – ComputerProgramming
c. *Required, elective, or selectedelective:*

***Specific goals for the course*:**

a. *Specific outcomes ofinstruction*:

1. The student will have a working knowledge of Internet Programming theory andpractice.
2. The student will design and experiment with various Internet Programmingconceptsandcomponentsviaprojects,toincreaseoverall understanding of modern InternetProgramming.
3. The student will be able to write and debug small distributed Java programs.

b. *ABET Criterion 3 Student Outcomes addressed by thecourse*:

(A): An ability to apply knowledge of mathematics, computing, science, and engineering appropriate to the discipline;

Students apply knowledge of computing and design to a project.

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

Projects are implemented in teams.

(I): An ability to use the current techniques, skills, and tools necessary for computing practice.;

Projects use current computing and modeling/design tools.

(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.*;*

Students are required to apply their knowledge of computing to design a solution to a problem and to document the solution including the tradeoffs involved in their design choices

Topics covered:

- Admin +Introduction
- Brief Introduction toJava
- Threads/Synchronization
- Basic NetworkConcepts
- Remote Invocation Method(RMI)
- SocketProgramming

6001214-3: Computer Organization & Assembly

Credits and contact hours: 3 Credits (2 30-minutes lectures perweek)

Instructor: Mona Alzhrani

Textbook: Computer Organization & Design: The Hardware/Software Interface, D. Patterson and J. Hennessy (3rd edition or newer)

***Specific courseinformation*:**

f. Instruction set architecture and MIPS assembly language, processor computation (data path and control), processor communication (cache and I/O modules).

g. *Prerequisite*: 6001213-3 – Logic Design & Analysis

h. Required, elective, or selected elective:

***Specific goals for the course*:**

e. *Specific outcomes of instruction*:

a. Understanding of instruction set architecture (ISA) and basic assembly language programming skills (MIPS ISA).

b. Understanding of processor computation by building processor data path and control.

c. Understanding of processor communication by cache memory and I/O modules.

f. *ABET Criterion 3 Student Outcomes addressed by the course*:

(A): An ability to apply knowledge of mathematics, computing, science, and engineering appropriate to the discipline;

Students apply knowledge of computing and design to a project.

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

Students are required design and implement a software project to meet a specification.

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

Projects are implemented in teams.

(F): An ability to communicate effectively;

The projects require communications, specifications, progress reports, and final report.

(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

Students are required to apply their knowledge of computing to design a solution to a problem and to document the solution including the tradeoffs involved in their design choices

Topics covered:

- Review of pre-requisites and introduction to computer organization.
- Instruction set architecture and MIPS assembly language.
- Processor data path and control.
- Exception and protection mechanisms.
- Memory hierarchies and cache.
- An overview of I/O modules and devices.

6001212-3 - Computers & Society (3 credits)

Credits and contact hours: 3 Credits (2 30-minutes lectures per week)

Instructor: Omar R. Alzubi

Textbook A Gift of Fire: Social, Legal, and Ethical Issues for Computers and the Internet (3rd Edition) by Sara Base

Specific course information:
This course explores basic cultural, social, legal, and ethical issues inherent in the

a. *Discipline of computing. Students will investigate important non-technical aspects of their role as a computing expert such as personal responsibility in ensuring faulty products are not released to market. Finally, students will see the importance of remaining up to date in their specialties and in computing as a whole, not just for personal benefit, but for society,too.*
b. *Prerequisite:No*
c. *Required, elective, or selectedelective:*

Specific goals for thecourse:

a. *Specific outcomes ofinstruction:*
b. *Students will be aware of, and be able to identify, the social, ethical,legal, professional, and privacy issues related tocomputing*
c. *Students will be able to articulate varying perspectives regardingethical, social, and professional issues in computer science andengineering*

Students will gain an appreciation for remaining up to date in their specialties

d. *ABET Criterion 3 Student Outcomes addressed by thecourse*:

(E). an understanding of professional, ethical, legal and social issues andresponsibilities.

A significant portion of course will be spent on educating students about professional, ethical, legal and social issues and responsibilities of a computing professional

(F). an ability to communicate effectively.

Students are required design and implement a software project to meet a specification.

(G). an ability to analyze the local and global impact of computingonindividuals,organizations and society, including ethical, legal, security and global policyissues.

Studentswillworkinteamsandanalyzetheimpactofachosencomputingtechnologyonindividuals, organizations, and society, including ethical, legal, security, and global policyissues.

(H). a recognition of the need for, and an ability to engage continuingprofessional Development.

The students will utilize the internet to search for examples of ethical, legal, and social impact of computing, and build an awareness for the need to stay aware of such issues in the future.

Topics covered:

- Social (cultural, international, govt) implications of computing, andinternet
- Identifying and evaluating ethical choices in softwaredesign
- Professionalism (care, attention, responsibility). Importance of keeping up todate.
- Codes of ethics, maintaining awareness of ethical consequences, ethicaldissent
- Historical examples of software risks (such as the Therac-25case)
- Computing in the workplaceissues
- Implications of softwarecomplexity
- Risk assessment and risk management; risk removal, risk reduction and riskcontrol
- Foundations of intellectual property (copyrights,patents)
- Softwarepiracy
- Ethical and legal basis for privacyprotection
- Privacy implications of database systems (e.g. data gathering, storage, and sharing) Technological strategies for privacyprotection

6001105-3 - Advanced Programming

Credits and contact hours: 3 Credits (2 x 50 mins lectures, 3 x 50 mins labs)

Instructor: Ismail Farahat

Textbook:

- **"Advanced programming (Java)", ThirdEdition.**
- **Design Patterns: Elements of Reusable Object-Oriented Software Gamma et al., Addison Wesley, 1994, ISBN0201633612**

***Specific course information*:**

a. Students will experience in advanced programming using the Java language, which The course aims to provide students with advanced skills in programmingdepending onwhathehadlearnedthebasicsofprogramming,tobeabletolinktheprogramsof healthdataenteredtothesystemandwordprocessingandidentificationoferrorsand exceptional treatment and the definition of new varieties to deal with exceptional errors.Studentswillalsoadvancetheirprogrammingskillsbyexploringgooddesign principles(e.g.,patterns),understandingbaddesignmethods(e.g.,anti-patterns),and gain early experience in O.O.P such as design, specification, andimplementation

b. *Prerequisite*: 6001104-3 - StructuredProgramming

c. *Required, elective, or selectedelective:*

***Specific goals for the course*:**

a. *Specific outcomes ofinstruction*:
   a. Students will know about the issues related to large-scale software development in O.O.P by Javalanguage
   b. Students will become better developers through knowledge of design patterns andanti-patterns.
   c. Students will gain early experience in O.O.P such as design, specificationand implementation

b. *ABET Criterion 3 Student Outcomes addressed by thecourse*:

(A): An ability to apply knowledge of mathematics, computing, science, and engineering appropriate to the discipline;

Students apply knowledge of computing and design to a project.

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

Students are required design and implement a software project to meet a specification.

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

Projects are implemented in teams.

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

Projects use current computing and modeling/design tools.

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

The students are required to use standard design and development principles on a significant software project.

Topics covered:

- Introductionofthecourseandrulesdealinginlectures,areviewofthebasicconceptsof programming
- WhatareAgvhprogramming?,Examplesofthefactit?,Whatisthedifferencebetweenthemand other programmingmethods?
- What is the class(Class)?
- And what is the object(Object)?
- How to deal with them through the Javalanguage?
- What are the Accessmodifier?
- Theapplicationofanumberofprogramstoexplainandgiveanopportunityforstudenttraining.
- Provide a range of applications PMS different levels of programmingaimed at the students themselvesBermojtahathentheteachercorrectionandcommentonthemostmistakes.
- And of these applicationsinclude:
- How Array ofobjects
- Application of the "Book of Walt Ifunat" through the goalprogramming
- Mid-termexam(practical)andbethefirstinthelastlectureoftheseventhweek.
- This period concentrated on explaining the concept of inheritance Inheritance and test how well the student has to absorb andunderstand.
- Explain the impact of inheritance on the AccessModifiers
- Explaintheconcept(Overloading)andSer(Overridden)withanexplanationofthedifference betweenthem.
- At the end of the tenth week (mid-termexam)
- Explain the concept of polymorphism Polymorphism, and how important itis.
- And explain how to deal with the application through the Java language, application of this concept through one or more applications and then give students the opportunity to interact with this concept.
- Review and discuss the students'projects

6001414-3: Natural Language Processing

Credits and contact hours: 3 Credits (2 30-minutes lectures perweek)

Instructor: HamzaAwadElkarimHamza

Textbook: Speech and Language Processing, 2nd Edition, Daniel Jurafsky and James H. Martin, Pearson

***Specific courseinformation*:**

a. *This course introduces students to concepts and ideas in natural language processing. It covers both the algorithms available for the processing of linguistic information and the fundamental computational properties of languages.*
b. *Prerequisite*: 6001332-3 – Intro to Artificial Intelligence

***Specific goals for thecourse*:**

Specific outcomes ofinstruction:

a. *Understanding* the role of regular grammars and CFGs in parsing different elements of a text corpus.
b. Identifying algorithms for different language processing tasks.
c. Study of different semantic models for natural languages.

ABET Criterion 2 Student Outcomes addressed by thecourse:

(A): An ability to apply knowledge of mathematics, computing, science, and engineering appropriate to the discipline;

Students apply knowledge of computing and design to a project.

(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.

Students are required to apply their knowledge of computing to design a solution to a problem and to document the solution including the tradeoffs involved in their design choices

Topics covered:

- Overview of NLP. Language models and their role in speech
- Processing Words and transducers
- N-gram Language Models and Information Theory
- Part-of-speech tagging, segmentation
- Syntactic parsing
- Statistical parsing
- Semantics
- Information Retrieval
- An Application in an Natural Language (Arabic, English)

6001330-3 - Computer Graphics (3 credits)

Credits and contact hours: 3 Credits (2 hours lecture and 3 hour Lab. per week)

Instructor: Amin Daoud

Textbook: Computer Graphics with OpenGL, Hearn & Baker, Prentice Hall OpenGL Programming Guide, Shreiner&Khronos OpenGL ARB Working Group, Addison- Wesley

***Specific course information*:**

a. The course offers an introduction to computer graphics hardware, algorithms, and software. Topics include overview of graphics hardware, 2D and 3D object representation, geometric transformations, 2D viewing, 3D viewing, illumination models, color modes, and colorapplications.

b. *Prerequisite*: 6001105-3 – AdvancedProgramming

c. *Required, elective, or selectedelective:*

***Specific goals for the course*:**

a. *Specific outcomes ofinstruction*:

a. The student will have a working knowledge of established software Understand the foundation of the implementation of computergraphics modeling and renderingsystems

b. Understand the mathematical background of computergraphics

c. Understand the handling ofcolors

d. Implementation of a graphics programmingproject.

b. *ABET Criterion 3 Student Outcomes addressed by thecourse*:

(A): an ability to apply knowledge of mathematics, computing, science, and engineering appropriate to the discipline;

Students apply knowledge of computer graphics (modeling and rendering) to complete assessments.

(C): an ability to design, implement and evaluate a computer-based system, process, component or program to meet desired;

Students design and write simple programs in labs. Students design and implement a software project to meet a specification.

(I): an ability to use the current techniques, skills, and tools necessary for computing practice;

Students use current computing and modeling/design tools such as OpenGL, Blender, etc.

(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*;*

Students use mathematical knowledge (vectors, transformations, modeling, etc.) to design a solution to a problem and to document the solution including the tradeoffs involved in their design choices

Topics covered:

.

- Introduction to computergraphics
- Computer graphicshardware
- Introduction toOpenGL
- Math for Computer Graphics (Trigonometry, Vectors, Projections, Interpolation)
- 2D objects drawing (usingOpenGL
- Line drawingalgorithms
- Matrices and 2DTransformations
- 3DTransformations
- 2D & 3DViewing
- Colormodels
- Animation
- Lighting

6001216-3: Multimedia Systems

Credits and contact hours: 3 Credits (3*50 minutes lectures perweek, 0 lab hour)

Instructor: SaadAlbogami

Textbook: Digital Multimedia, Chapman & Chapman, ISBN 978-0-470-51216-6.

***Specific courseinformation*:**

i. A course on theoretical and practical issues in designing multimedia systems. Topics will include digital video coding, Scalable Video Coding, content protection (watermarking), and design of multimedia middleware (e.g., multimedia authoring). Standards such as MPEG-2, MPEG-4, H.264, MPEG-7, and MPEG-21 will also be reviewed.

j. *Prerequisite*: 1401211-3 – Web Programming
k. Required, elective, or selected elective: None

***Specific goals for thecourse*:**

g. *Specific outcomes ofinstruction*:
   a. Identify different multimedia data types such as image, audio, and video.
   b. Understand basics of digital audio representation.
   c. Understand basic image data representations.
   d. Learn fundamental concepts in video.
   e. Learn how to capture, digitize, store, and manipulate a variety of multimedia data.
   f. Use some multimedia authoring tools and system.
   g. Familiarize with basic audio, image, and video coding/compression techniques such as MPEG, JPEG and some advanced topics in multimedia.

h. *ABET Criterion 3 Student Outcomes addressed by thecourse*:

(A): An ability to apply knowledge of mathematics, computing, science, and engineering appropriate to the discipline;
Students are required to apply knowledge of mathematics to understand digitizing sound and images, conversion between color models and compression techniques
(E): an understanding of professional, ethical, legal and social issues and responsibilities
Students are required to use professional software used for editing multimedia elements.
(F): An ability to communicate effectively
Group presentation allowed to student to present their ideas and communicate them effectively.
(H): Arecognition of the need for, and an ability to engage continuing professional development ability to communicate effectively.
Students are required to search internet to find suitable authoring tools for their project.
(I): An ability to use the current techniques, skills, and tools necessary for computing practice.
Students are required to use the latest authoring software for their projects.

(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.

Students are required to understand the different compression techniques used in image and video.

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

The students are required to utilize multimedia authoring tools to create an interactive multimedia project.

Topics covered:

- Introduction to Multimedia.
- Graphic, Images and Color.
- Audio Technology.
- Multimedia Authoring.
- Video Technology, Computer-based Animation Data Compression: Image, Video and Audio Standards.
- Data Compression: MPEG Video Coding.
- Advanced Topics: Content based retrieval, Media servers, Synchronization, P2P and VOIP.

6001215-3 - Computer Architecture

Credits and contact hours: 3 Credits (3 x 50 mins lectures, 0 lab hours)

Instructor:

Textbook:

- *Computer* **Organization & Design: The Hardware/Software Interface, D. Patterson and J. Hennessy (3rd edition or newer)**

***Specific courseinformation*:**

a. This course extends computer organization course by covering advanced processor features that are standard in modern processors, and exploring the design and trade-offs of memory hierarchies, including implications for parallel processor architectures.
b. Prerequisite 6001214-3- Computer Organization & Assembly Language
c. Required, elective*, or selected elective:*

***Specific goals for thecourse*:**

a. *Specific* outcomes *ofinstruction*:
    a. Understanding of computer performance evaluation
    b. Understanding of pipelined CPU and cache hierarchy
    c. Understand fundamental concepts in multiprocessing
b. *ABET Criterion 3 Student Outcomes addressed by thecourse*:

(A): An ability to apply knowledge of mathematics, computing, science, and engineering appropriate to the discipline;

Students apply knowledge of computing and design to a project .

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

The course provides students of assessing different engineering solutions in terms of architecture speed (MIPS), cost and their trade off.

(H): A recognition of the need for, and an ability to engage continuing professional development;

The final paper aims at recognizing the need to read about new architectures and organizations in a field that advances at a high pace

(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.;

Students are required to apply their knowledge of computing to design a small procedural program.

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

The students are required to use standard design and development principles on a significant software project .

Topics covered:

- Review of pre-requisites and introduction to computer architecture
- Evaluation of computer performance
- Enhancing computer performance with pipelining (data path and control)
- Data Hazards, forwarding, stalls, branch hazards, exceptions
- Exploitation of memory hierarchy (Cache and virtual memory)
- Fundamentals of multiprocessing (shared memory, clusters and message passing)
- Hardware multithreading, SISD, MIMD, SIMD, SPMD and vector processing
- Fundamentals of instruction level parallelism

6001335-3: Advanced Databases

Credits and contact hours: 3 Credits (3 x 50 mins lectures, 0 lab hours)

Instructor:

Textbook: Pearson International Edition, ISBN: 0132144980, 2011.

***Specific courseinformation*:**

a. *The* enhanced entity-relationship (EER) model. Relational database design by ER- and EER-to-relational mapping. Concepts for object databases. Object database standards, languages, and design. Object-relational databases. XML databases. Database transaction and query processing. Distributed databases. Database security. Database tuning and recovery.
b. *Prerequisite*: *6001312-3* - Fundamentals of databases
c. Required, elective, or selected elective: None

***Specific goals for thecourse*:**

a. *Specific outcomes ofinstruction*:
    a. The students will understand different terms of advanced data modeling e.g. object, object-relational, and XML and the supporting theoretical foundation.
    b. The students will learn techniques of advanced schema mapping i.e. from enhanced entity relation to relational, object to relational, object-relational to relational, and xml to relational.
    c. The students will understand advanced database topics such as indexing, query processing, local and distributed transaction processing, and security.
b. *ABET Criterion 3 Student Outcomes addressed by thecourse*:

(A): An ability to apply knowledge of mathematics, computing, science, and engineering appropriate to the discipline;
Students apply knowledge of computing and design to a project
(C): An ability to design, implement and evaluate a computer-based system, process, component or program to meet desired;
Students are required to design and implement a software project to meet a specification.
(D): An ability to function effectively on teams to accomplish a common goal
Projects are implemented in teams.
(H): a recognition of the need for, and an ability to engage continuing professional
The students often must utilize database vendors blogs and open source sites to learn and apply the new technologies that they have chosen in support of their projects.
(I): An ability to use the current techniques, skills, and tools necessary for computing practice.
Projects use current computing and modeling/design tools.

(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.;

Students are required to apply their knowledge of computing to design a solution to a problem and to document the solution including the tradeoffs involved in their design choices.

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

The students are required to use standard design and development principles on a significant software project

Topics covered:

- Relational algebra: relations, tuples, attributes, schemas, relational operators and expressions.
- Functional Dependencies: keys, closures, Armstrong's axioms, canonical cover.
- Normalization: anomalies, lossless decomposition, dependency preservation, BCNF, 3NF.
- SQL queries: types, 3 valued logic, nulls, select, ordering, joins, set operators, aggregate functions, grouping, sub-queries.
- SQL data manipulation: insertion, deletion, and update.
- SQL data definition: schema definition, default, primary key, unique, not null, check, assertions, foreign keys, referential integrity, views.
- Transactions: failures, atomicity, consistency, isolation, durability.
- Entity Relationship Modeling: Entities, relationships, attributes, ER diagrams, relationships, participation, fan and chasm traps, roles, weak entities, mapping to relation schemas, is-a relationships and hierarchies.
- Database Indexing

6001218-4: Data Structures & Algorithms

Credits and contact hours: 4 Credits (2 x 50 mins lectures, 2 x 50 mins labs)

Instructor: HamzaAwadelkarimHamza Ibrahim

Textbook: Data Structures and Algorithms in Java, 4th edition, by M.T. Goodrich and R. Tamassia. John Wiley and Sons, Inc., ISBN: 0-471-73884-0

Specific courseinformation:

a. A data *structure* is the logical arrangement of data elements, combined with the set of operations we need to access the elements.The objective of this course is to provide the fundamentals of data structures and algorithm design needed in the remainder of the curriculum, introduce algorithm analysis, and develop students' problem solving and computer programming skills. Emphasis on linked lists, stacks, queues, trees, priority queues, heaps and graphs, and abstract data types. Also includes object oriented concepts.

b. *Prerequisite*: *6001105-3* – Advanced Programming

Specific goals for thecourse:

a. *Specific outcomes ofinstruction*:
   a. The student willbe able to describe, construct, and use various implementations for fundamental data abstractions such as lists, stacks, queues, trees, and graphs.
   b. The student will be able to design and implement efficient algorithms for manipulating data structures.
   c. The student will be able to compare the efficiency of various data structures and algorithms and to choose the most appropriate ones for a given application.

b. *ABET Criterion 3 Student Outcomes addressed by thecourse*:

(A): An ability to apply knowledge of mathematics, computing, science, and engineering appropriate to the discipline;
Students apply knowledge of computing and design to a project.
(B): An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution
On theoretical level students will be able to see which problems can be solved on which model.
(C): An ability to design, implement and evaluate a computer-based system, process, component or program to meet desired;
Students are required design and implement a software project to meet a specification.
(I): An ability to use the current techniques, skills, and tools necessary for computing practice.
Projects use current computing and modeling/design tools.

(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.;

Students are required to apply their knowledge of computing to design a solution to a problem and to document the solution including the tradeoffs involved in their design choices

Topics covered:

- Basics of algorithm analysis
- Linear Data Structures
- Sorting
- Search Trees
- Hash Tables
- Priority Queues
- Graphs

6001231-4: Programming Languages

Credits and contact hours: 4 Credits (450-minutes lectures perweek)

Instructor: ManalAlasmari

Textbook: Sebesta R.W., Concepts of Programming Languages, 9th Edition, Addison-Wesley, 2010

***Specific course information*:**

c. An introduction to programming language, specification and analysis. Additional topics include control structures, data types and structures, runtime, environments, binding strategies, compilers, and interpreters.

a. *Prerequisite*: 6001105-3 - Advanced Programming ,6001217-3 - Logic Programming
b. Required, elective, or selected elective:

***Specific goals for the course*:**

d. *Specific outcomes of instruction*:

a. Apply concepts from prerequisite courses, especially formal languages and architecture courses, in the context of evaluating the features of programming languages.
b. Explain and evaluate design and implementation features of programming languages.
c. Apply conceptual knowledge of the syntax of languages, as well as the design of language data structures and control statements, to the efficient implementation of a working language.

e. *ABET Criterion 3 Student Outcomes addressed by the course*:

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

Students could determine the language that is suitable language for programming each problem

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

Projects are implemented in teams.

(F): An ability to communicate effectively;

The projects require communications, specifications, progress reports, and final report.

(H): A recognition of the need for, and an ability to engage continuing professional development;

The students often must utilize the internet to learn and apply the new technologies that they have chosen in support of their projects.

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

Projects use current computing and modeling/design tools.

Topics covered:

- Preliminaries
- Evolution of the Major Programming Languages
- Describing Syntax and Semantics
- Names, Variables, Bindings and Type Checking.
- Scope and lifetime.
- Referencing Environments, Named Constants
- Primitive Data Types, Character String Types
- User-Defined Ordinal Types
- Array Types and Associative Arrays
- Record Types and Union Types
- Pointer and Reference Types

6001315-3 - Computer Theory

Credits and contact hours: 3 Credits (2h 30-minutes lectures per week)

Instructor: Mohamed Amin HADJ TAIEB

Textbook:

- MS] Introduction to the Theory of Computation, 2nd Edition, By MichaelSipser
- [HMU] Introduction to Automata Theory, Languages, and Computation,3rd Edition, By J. Hopcroft, R. Motwani, J.Ullman

***Specific course information*:**

a. Theaimofthiscourseistopresentthekeyconceptsoftheoryofcomputation, atopicthataffectsallcomputersciencecourses.Themaintopicsofthecourse are languages, grammars, and automata, graph construction from grammars and automata, and complexitytheory.

b. Prerequisite:

6001227-3 – File Processing and Organization

6001231-4 – Programming Languages

c. Required, elective, or selectedelective:

***Specific goals for the course*:**

a. *Specific outcomes ofinstruction*:

a. Develop skills in formal and precise reasoning in the field of computer science.
b. Understand formal definitions of machine models, grammars and languages and the concepts of determinism andnon-determinism.
c. Understand the theoretical limits of different computationalmodels.

b. *ABET Criterion 4 Student Outcomes addressed by thecourse*:

(A): An ability to apply knowledge of mathematics, computing, science, and engineering appropriate to the discipline;

Students apply knowledge of computing and design to a project.

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

On theoretical level students will be able to see which problems can be solved on which model.

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

Students will be able to model computational problems using finite state machines and push down automata.

(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.

Students are required to apply their knowledge of computing to design a solution to a problem and to document the solution including the tradeoffs involved in their design choices

Topics covered:

- Admin +Introduction
- Automata, Computability andComplexity
- Finite Automata(DFA)
- Non deterministic automata(NFA)
- Regular expression, Regularlanguages
- Nonregular languages, PumpingLemma
- Context-free Grammar andLanguages
- Parse Tree,Ambiguity
- Pushdown Automata(PDA)
- Equivalence of PDA andCFG
- Non-context-free Languages, PumpingLemma
- Turingmachines
- Decidability andcomplexity

6001311-3: Operating Systems

Credits and contact hours: 4 Credits (3*50 minutes lectures perweek, 0 lab hour)

Instructor: Abdelrahman Osman

Textbook: Operating Systems Concepts, seventh edition, Silberchatz, Galvin, and Gagne, John Wiley & Sons Inc., ISBN 0-471-69466-5.

***Specific courseinformation*:**

l. This course provides an introduction to operating system design and implementation. It covers the major components of most operating systems, in particular process management, memory management (segmentation, paging, swapping), file systems, and OS protection and security.

m. *Prerequisite*: 6001215-3 - Computer Architecture

n. Required, elective, or selected elective: None

***Specific goals for thecourse*:**

i. *Specific outcomes ofinstruction*:

a. Awareness of basic components of operating system and knowledge of the services provided by it.
b. Appreciate the main principles and techniques used to implement processes and threads, inter-process communication, process synchronization, and algorithms for process scheduling.
c. Appreciate memory management techniques including virtual memory abstractions.
d. Appreciate I/O mechanisms, disk organization and file system structure.
e. Evaluate security risks in operating systems and understand the role operating systems can and should play in establishing security.

j. *ABET Criterion 3 Student Outcomes addressed by thecourse*:

(A): An ability to apply knowledge of mathematics, computing, science, and engineering appropriate to the discipline;

Students apply knowledge of computing and design to programming assignments

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

Programming assignments are implemented in teams.

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

Students leverage the capabilities of a modern OS to solve real problems.

(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.

Students are required to apply their knowledge of computing to design a solution to a problem and to document the solution including the tradeoffs involved in their design choices.

Topics covered:

- Introduction to Operating Systems
- Process and thread management
- Memory management
- File system
- I/O system
- Protection & Security

6001310-3- Compiler Construction

Credits and contact hours: 3 Credits (3,0,0) hours lectures perweek)

Instructor: AbdelkarimAbdelkader

Textbook: Compilers: Principles, Techniques, and Tools ,A. V. Aho, R. Sethi, J. D. Ullman; (c) 2010;

***Specific courseinformation*:**

o. Compiler construction: lexical analysis, including regular languages and finite-state acceptors; syntactic analysis, including parsing techniques and grammars; code generation and optimization.

p. *Prerequisite*: 6001231-4 Programming Languages
q. Required, elective, or selected elective: Required

***Specific goals for thecourse*:**

k. *Specific outcomes ofinstruction*:
   a. Understanding of the organization of a compiler
   b. Understanding of the concepts of scanning, parsing, and translation
   c. Understanding of Compiler writing tools.
l. *ABET Criterion 3 Student Outcomes addressed by thecourse*:

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

Students are required to implement a simple compiler to translate infix code representation to postfix representation.

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

The assigned project and assignments and presentation at the end of course enable students to communicate effectively.

(I): An ability to use the current techniques, skills, and tools necessary for computing practice.;

Projects use current computing and modeling/design tools .

Topics covered:

- Introduction to compilers structure & goals
- Arithmetic expression processing using a stack
- Simple compiler structure

- Grammar, parse tree, and ambiguous grammar
- Translation schemes
- Context-free grammar & parsing
- Introduction to left recursion and right recursion
- Lexical analyzer (language, errors, pattern specifications)
- Operations on languages and regular expressions
- Finite automata
- Parsers and errors and sentential error
- Left recursion and left factoring
- FIRST, FOLLOW, and transition diagrams

6001313-3: Software Engineering

Credits and contact hours: 3 Credits (2 30-minutes lectures per week)

Instructor: Ismail Farahat

Textbook: Software Engineering – Principle and Practice Hans Van Vliet, 3rd, 2010, 978-0-470-03146

***Specific course information*:**

a. Software engineering concepts including the software life cycle and other software-development process models. Specification techniques, design methodologies, performance analysis, and verification techniques. Team-oriented software design and development, and project management techniques. Introduction to design and debugging tools of a modern programming language. Homework and laboratory projects that emphasize design and the use/features of a modern programming language in software development.

b. *Prerequisite*: 6001222-3 – System Analysis &Design

c. *Required, elective, or selectedelective:*

***Specific goals for the course*:**

a. *Specific outcomes ofinstruction*:
   a. The student will have a working knowledge of established software engineering issues and practice and their relationship to emerging methodologies, paradigms, techniques, tools, andlanguages.
   b. The student will be able to analyze, design and implement a modern application from an architectural perspective, which includes a decomposition into components of software, hardware, and their interdependencies.
   c. The student will be able to design and prototype software from written specifications and/or supplied applicationlibraries.

b. *ABET Criterion 3 Student Outcomes addressed by thecourse*:

(A): An ability to apply knowledge of mathematics, computing, science, and engineering appropriate to the discipline;
Students apply knowledge of computing and design to a project.

(C): An ability to design, implement and evaluate a computer-based system, process, component or program to meet desired;
Students are required design and implement a software project to meet a specification.

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

Projects are implemented in teams.

(F): An ability to communicate effectively;

The projects require communications, specifications, progress reports, and final report.

(H): A recognition of the need for, and an ability to engage continuing professional development;

The students often must utilize the internet to learn and apply the new technologies that they have chosen in support of their projects.

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

Projects use current computing and modeling/design tools.

(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.;

Students are required to apply their knowledge of computing to design a solution to a problem and to document the solution including the tradeoffs involved in their design choices

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

The students are required to use standard design and development principles on a significant software project.

Topics covered:

- Introduction to software engineering and its impact on softwaredevelopment.
- Critical software engineering principles such as modularity, abstraction, software evolution,etc.
- Software development process models such as waterfall, spiral, etc. and case studies on their usage. Object-oriented developmentmodels.
- Traditional and object-oriented software design concepts andtechniques.
- Software verification via testing, analysis, anddebugging.
- Software engineering tools and environments. Practice in using tools forsoftware design, and testing.
- Basic management concepts including an introduction to team aspects of solving software designproblems.

6001334-3 - Parallel & Distributed Computer Systems

Credits and contact hours: 3 Credits (3 x 50 mins lectures, 0 lab hours)

Instructor: GaaferWadidi

Textbook:

- *Introduction to Parallel Programming, Peter Pacheco, 2011*

***Specific courseinformation*:**

a. Introduction to parallel computing using shared memory and distributed memory multi-core computers, including hands-on practice with such systems during programming homework assignments, and a team project.
b. Prerequisite 6001311-3 – Operating Systems
6001312-3 – Fundamentals of Databases
c. *Required,* elective*, or selected elective:*

***Specific goals for thecourse*:**

a. Specific outcomes of instruction:

1. *Awareness of basic multiprocessor hardware taxonomy*
2. *A strong grasp of the basic software and hardware strategies for managing access to shared data (from locks, and barriers, to cache coherency)*
3. *An in-depth understanding of the major sources of performance loss in parallel programs, and some general solutions to reducing performance loss*
4. *Ability to use standard parallel programming APIs such as Pthreads, OpenMP, and MPI through practice on shared memory and distributed memory computers*

(A): An ability to apply knowledge of mathematics, computing, science, and engineering appropriate to the discipline;

Students apply knowledge of computing and design to a project .

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

Students learn the basics of gathering requirements for solving a problem

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

Students are required design and implement a software project to meet a specification .

(I): An ability to use the current techniques, skills, and tools necessary for computing practice.;

Projects use current computing and modeling/design tools .

(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.;

Students are required to apply their knowledge of computing to design a small procedural program.

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

The students are required to use standard design and development principles on a significant software project .

Topics covered:

- Multi-core and its implications on software engineers
- Parallel architectures
- Source of loss in parallel performance
- Accessing shared data safely
- General parallel algorithmic models
- Pthreads, OpenMP, and MPILists and String in Prolog.

6001333-3: Human Computer Interaction

Credits and contact hours: 3 Credits (1.20 & 60 minutes lectures per week)

Instructor: Marwan Al-Namari

Textbook: Designing the User Interface: Strategies for Effective Human-Computer Interaction, 5/E, Shneiderman et al., ISBN: 0321537351, Pearson, 2009

***Specific course information*:**

a. Studentswillgainanunderstandingofuserinterfacedesign,andalternatives totraditional"keyboardandmouse"computing,includingvirtualreality,and ubiquitous computing. Students will become familiar with sensory and cognitive systems and be able to apply models from cognitive psychology to predicting user performance in various human-computer interaction tasks and recognize the limits of human performance as they apply to computer operation. Students will appreciate the importance of a design and evaluation methodology that begins with and maintains a focus on the user, thesocialimplicationsoftechnologyandethicalresponsibilitiesinthe design of technologicalsystems.

b. *Prerequisite*: 6001216-3 – MultimediaSystems
c. *Required, elective, or selectedelective:*

***Specific goals for the course*:**

a. *Specific outcomes ofinstruction*:
    a. The Students will know key concepts in designing usableproducts.
    b. Students will be able to evaluate the usability of a given computer- basedsolution

b. *ABET Criterion 2 Student Outcomes addressed by thecourse*:

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

Students are required design and implement a software project to meet a specification.

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

Projects use current computing and modeling/design tools.

Topics covered:

- Background--the development, context, and scope ofHCI
- User-Centered Design, Understanding and ObservingUsers
- Use Case Scenarios, Personas, and UserModeling
- User Experience, Usability Requirements, and Low-FidelityPrototyping
- Usability Testing, Interaction Design and AnalyticalEvaluation
- Models and Theories: GOMS, MHP, Fitts'Law
- Accessibility and Risks: Error Classification, Automation, Designing forError
- Graphics and Sound: GUIs, speech and non-speechaudio
- Multimodal Interfaces and UbiquitousComputing
- Gestural Interaction and Tangible UserInterfaces
- Affective and SocialComputing
- Augmented Reality and Computer Supported CooperativeWork

6001332-3: Introduction to Artificial Intelligence

Credits and contact hours: 3 Credits (3 50-minutes lectures per week)

Instructor: Dr. MusabBassam Al-Zghoul

Textbook: Artificial Intelligence: A Modern Approach Russell &Norvig, 3rd, 2009, 0136042597.

***Specific course information*:**

a. Introduction to Artificial Intelligence introduces basics of Artificial Intelligence, concept of intelligent agents and various types of agents. It includesvarioussearchtechniques,Propositionallogicandfirstorderlogic.It further introduces the concept of knowledge engineering and inference systems.

b. Prerequisite: 6001315-3 – ComputerTheory

c. Required, elective, or selectedelective:

***Specific goals for the course*:**

c. *Specific outcomes ofinstruction*:

a. Students will learn basics of AI, Intelligent Agents and their different typesandapplications.

b. They will learn in detail different search techniques including uninformed search, heuristic search, adversarial search that can be used in Game playing and other AIapplications.

c. Students will learn logical agents, first order logic and first order inferencesystem.

d. *ABET Criterion 3 Student Outcomes addressed by thecourse*:

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

Students are required design and implement a software project to meet a specification.

(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.*;*

Students are required to apply their knowledge of computing to design a solution to a problem and to document the solution including the tradeoffs involved in their design choices

***Topics covered*:**

- Introduction toAI.
- History ofAI.

- AIDomains.
- AIApplications.
- Intelligent Agents andEnvironment.
- Structure of Different type ofAgents.
- Problem Solving through search Un-informed search (BFS, DFS, Depth First, Depth limited and iterative deepeningsearch).
- Informed Search (Greedy best first search, A* search,Heuristics).
- Local Search Algorithms (Hill Climbing, SimulatedAnnealing).
- Adversarial Search (Minimax Algorithm, Alpha Beta Pruning, ChanceMinimax).
- Logical Agents (knowledge based agents, propositional logic, First OrderLogic, Knowledge.
- Engineering inFOL.
- Inference inFOL

6001419-4: Research Project

Credits and contact hours: 4 Credits (4*50 minutes lectures perweek, 0 lab hour)

Instructor: AbdelkarimAbdelkader

Textbook: UQU Undergraduate Final Year Project Handbook.

***Specific courseinformation*:**

r. This course is the first semester of the required major design experience. In a two semester-long project, student teams will propose, design, produce and evaluate a software and/or hardware system. The project will culminate in the delivery of a working system, a formal public presentation, and written documentation. Oral and written progress reports are required.

s. *Prerequisite*: 6001311-3 Operating Systems - 6001335-3 Advanced Database Systems

t. Required, elective, or selected elective: None

***Specific goals for thecourse*:**

m. *Specific outcomes ofinstruction*:

a. Ability to apply knowledge of mathematics, computing, science, and engineering appropriate to the discipline.
b. Ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.
c. Ability to design, implement and evaluate a computer-based system, process, component or program to meet desired goal.
d. Ability to function effectively on teams to accomplish a common goal.
e. Ability to communicate effectively.
f. Recognition of the need for, and an ability to engage continuing professional development.
g. Ability to use the current techniques, skills, and tools necessary for computing practice.
h. Apply mathematical foundations, algorithmic principles, and computer science theory in the modelling and design of computer based systems in a way that demonstrates comprehension of the trade-offs involved in design choices.
i. Ability to apply design and development principles in the construction of software systems of varying complexity.

n. *ABET Criterion 3 Student Outcomes addressed by thecourse*:

(A): An ability to apply knowledge of mathematics, computing, science, and engineering appropriate to the discipline;
Students apply knowledge of computing and design to a project
(B): An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.
Students will develop project requirement specification
(D): An ability to function effectively on teams to accomplish a common goal

Projects are implemented in teams.

(F): An ability to communicate effectively

The projects require communications, specifications, progress reports, and final report.

(H): Arecognition of the need for, and an ability to engage continuing professional development ability to communicate effectively.

The students often must utilize the internet to learn and apply the new technologies that they have chosen in support of their projects.

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

Projects use current computing and modeling/design tools.

(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.

Students are required to apply their knowledge of computing to design a solution to a problem and to document the solution including the tradeoffs involved in their design choices.

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

The students are required to use standard design and development principles on a significant software project

Topics covered:

N/A

6001439-4: Graduation Project

Credits and contact hours: 4 Credits (4*50 minutes lectures perweek, 0 lab hour)

Instructor: AbdulkareemAbdulgader

Textbook: UQU Undergraduate Final Year Project Handbook.

***Specific courseinformation*:**

u. This course is the second semester of the required major design experience. In a two semester-long project, student teams will propose, design, produce and evaluate a software and/or hardware system. The project will culminate in the delivery of a working system, a formal public presentation, and written documentation. Oral and written progress reports are required.

v. *Prerequisite*: 6001419-4 – Research Project

w. Required, elective, or selected elective: None

***Specific goals for thecourse*:**

o. *Specific outcomes ofinstruction*:

a. Ability to apply knowledge of mathematics, computing, science, and engineering appropriate to the discipline.
b. Ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.
c. Ability to design, implement and evaluate a computer-based system, process, component or program to meet desired goal.
d. Ability to function effectively on teams to accomplish a common goal.
e. Ability to communicate effectively.
f. Recognition of the need for, and an ability to engage continuing professional development.
g. Ability to use the current techniques, skills, and tools necessary for computing practice.
h. Apply mathematical foundations, algorithmic principles, and computer science theory in the modelling and design of computer based systems in a way that demonstrates comprehension of the trade-offs involved in design choices.
i. Ability to apply design and development principles in the construction of software systems of varying complexity.

p. *ABET Criterion 3 Student Outcomes addressed by thecourse*:

(A): An ability to apply knowledge of mathematics, computing, science, and engineering appropriate to the discipline;
Students apply knowledge of computing and design to a project
(B): An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.
Students will develop project requirement specification

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

Students are required to design and implement a software project to meet a specification.

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

Projects are implemented in teams.

(F): An ability to communicate effectively

The projects require communications, specifications, progress reports, and final report.

(H): Arecognition of the need for, and an ability to engage continuing professional development ability to communicate effectively.

The students often must utilize the internet to learn and apply the new technologies that they have chosen in support of their projects.

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

Projects use current computing and modeling/design tools.

(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.

Students are required to apply their knowledge of computing to design a solution to a problem and to document the solution including the tradeoffs involved in their design choices.

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

The students are required to use standard design and development principles on a significant software project

Topics covered:

N/A

6001417-3: Computer Network Systems

Credits and contact hours: 3 Credits (3*50 minutes lectures per week, 0 lab hour)

Instructor: Hassan Alamri

Textbook: "Computer Networking: A Top-Down Approach", James Kurose and Keith Ross , 5th edition ISBN: 0136079679, Publisher: Addison-Wesley, 2009.

***Specific course information*:**

a. The course covers principles of computer networking with the focus on the Internet. Thestructure,practices,protocolsandcomponentsofcomputernetworksinvolvedin supporting the Internet, are studied in detail. Important concepts discussed in the course are related to packet switching, layered architecture, TCP/IP protocol suite, window flow control and local area networks. Simulations are used for visualization of network relatedconcepts.

b. *Prerequisite*: 6001311-3 – OperatingSystems
c. *Required, elective, or selectedelective:*

***Specific goals for the course*:**

a. *Specific outcomes ofinstruction*:
    a. Students will have a working knowledge of computer networks, Internetinparticular.Theywillbeabletodescribethetopicsandsolverelate dproblems.
    b. Students will have skills to develop network applications based on client-serverarchitecture.
    c. Studentwillhaveabilitytoapplymathematicalknowledgeforanalysis of networkprotocols.
    d. Student will have ability to identify, analyze and workout network engineeringproblems.
    e. Student will have ability to use networking tools for engineering practices.

b. *ABET Criterion 3 Student Outcomes addressed by thecourse*:

(A): An ability to apply knowledge of mathematics, computing, science, and engineering appropriate to the discipline;
Students are required to have a good understanding and knowledge of principles of networking to successfully pass all the evaluation components
(C): An ability to design, implement and evaluate a computer-based system, process, component or program to meet desired;
Students are required design and implement a software project to meet a specification.
(D): An ability to function effectively on teams to accomplish a commongoal

Projects are implemented in teams.

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

Students are required to use sophisticated network analyzer in labs to visualize working of different protocols on different network layers. Network Simulator is used to visualize network

(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.

Students are required to apply their knowledge of mathematics and computing to solve a number of networking problems. Moreover, the design of the project requires computing and algorithmic skills.

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

The students are required to use standard design and development principles on a significant software project.

Topics covered:

- Introduction to Computer Networks, Internet Architecture, Circuit and Packet Switching, AccessSystems.
- Application Layer Principles, HTTP, DNS, Peer to PeerNetworks.
- Transport Layer Services, UDP, Reliable data delivery, TCP, CongestionControl.
- Network Layer Services, Virtual Circuits, IP, Addressing, RoutingProtocols.
- LinkLayerServices,MultipleAccessProtocols,Linklayeraddressing,Ethernet,Switches.
- Introduction to Wireless and Mobile Networks, Wireless characteristics, CDMA, Cellular Networks,Mobility.

6001432-3: Computer Security Systems

Credits and contact hours: 3 Credits (3 50-minutes lectures per week)

Instructor: Dr. MusabBassam Al-Zghoul

Textbook: Computer Security: Principles and Practice William Stallings and Lawrie Brown, 2rd, 2010.

***Specific course information*:**

a. Computer Security Systems is the first level of computer and network security. The course will cover various topics related to computer security, data privacy, network protection against various attacks. The course gives students enough knowledge and a reasonable background to understand network security, active and passive attacks, Internet privacy, secure communications. Students are expected to practice biweekly homework, develop critical thinking about computer and network security, and apply learned materials in different contexts of various attacks, wireless and Internetsecurity.

b. *Prerequisite*: 6001311-3 – OperatingSystems

c. *Required, elective, or selectedelective:*

***Specific goals for the course*:**

a. *Specific outcomes ofinstruction*:
   a. Appreciate the need for computer security and computer protection, including the tradeoffs between different security and protection methods.
   b. Able to apply concepts of public keys, private keys, cryptosystem, authentication, digital signatures to secure simplesystems.
   c. Implement some network security protocols such as SSL, MAC, and wireless security, WEP, WAP, and computer viruses, and Internet attacks,andutilizetheminrealapplicationstosecureIntern ettraffic.

b. *ABET Criterion 3 Student Outcomes addressed by thecourse*:

(A): An ability to apply knowledge of mathematics, computing, science, and engineering appropriate to the discipline;
Students apply knowledge of computing and design to a project.

(E): an understanding of professional, ethical, legal and social issues and responsibilities;

Students will appreciate the implications of leaving systems insecure

(G): an ability to analyze the local and global impact of computing on individuals, organizations and society, including ethical, legal, security and global policy issues

Students will be able to analyze the impact of security on organizations and individuals

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

Projects use current computing and modeling/design tools.

Topics covered:

- Overview of computer securityservices.
- Passive and activeattacks.
- Cryptographic public and symmetric keys:DES.
- Advanced Encryption StandardAES.
- Public key cryptography, and RSAalgorithm.
- El-Gamalcryptosystem.
- Digital signatures and message authenticationprotocols.
- Transport layer security, SSL protocol, MACscheme.
- Wireless security protocols, WPA,WEP.
- Viruses, and Internetattacks.

6001430-3- Arabization of Computer Systems

Credits and contact hours: 3 Credits (2:30-minutes lectures perweek)

Instructor: Ismail Farahat

Textbook: Arabization of Computing –Dr. ahmed –abdelfattah-elharby

***Specific courseinformation*:**

x. This course introduces students to concepts and ideas in Arabization of Computer . It covers both Personalcreativeproducehasthe ability toself-produce industry,science andtechnology andthe Arab worldthusbecomesaArabizationColumns listby theArab-IslamicRenaissanceHence the"Arabization ofComputing" Arabizationas oneof the most importanttraffic hubs And clarify theconceptsbetween translationandArabization ofComputer.

y. *Prerequisite*: 6001414-3 – Natural Language Processing

z. Required, elective, or selected elective:

***Specific goals for thecourse*:**

q. *Specific outcomes ofinstruction*:
   a. Understanding the regular roles of Arabization and in parsing different elements of arabization.
   b. Identifying the requirement for Globalization.
   c. Study of different semantic models for Arabization .

r. *ABET Criterion 3 Student Outcomes addressed by thecourse*:

(A): An ability to apply knowledge of mathematics, computing, science, and engineering appropriate to the discipline;
Students apply knowledge of computing and design to a project .

(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.

Students are required to apply their knowledge of computing to design a solution to a problem and to document the solution including the tradeoffs involved in their design choices

Topics covered:

- Overview of Arabization models and characteristics of Arabic Language.
- History and Arabization of technical terminology.
- Arabization of technical terminology.

- Arabic operating systems and programming languages.
- Globalization.
- Proposal forprojects.
- Automaticreadingof the textwith spell checker.
- Automatic systemforthe writing of Arabicnamesin lettersEnglish.

6001413-3 - Expert Systems (3 credits)

Credits and contact hours: 3 Credits (2,5 hours lecture per week)

Instructor: **Amin Daoud**

Textbook: Expert Systems Principles and Programming (4th edition) By Joseph Giarratano and G. Riley. Published by PWS Publishing, Boston, MA, 2004

***Specific courseinformation*:**

aa. Knowledge Acquisition techniques, Knowledge representation, Analysis and Design of an ES, Reasoning strategies, Software tools and languages, Applications of ES, Validation and verification of ES..

bb. *Prerequisite*6001332-3 – Artificial Intelligence

cc. Required, elective, or selected elective:

***Specific goals for thecourse*:**

s. *Specific outcomes ofinstruction*:

1. To understand expert systems fundamentals including knowledge: types, engineering process activities and acquisition.
2. To understand and use a wide variety of representation and inference techniques to deal with the knowledge in Expert system.
3. Become familiar with the design and implementation of Expert System and rule base using CLIPS or any other language.
4. .Become familiar to Fuzzy Expert System.

t. *ABET Criterion 3 Student Outcomes addressed by thecourse*:

(A): an ability to apply knowledge of mathematics, computing, science, and engineering appropriate to the discipline;
Students apply knowledge of computer graphics (modeling and rendering) to complete assessments.
(C): an ability to design, implement and evaluate a computer-based system, process, component or program to meet desired;
Students design and write simple programs in labs. Students design and implement a software project to meet a specification.
(F) an ability to communicate effectively;
The projects require communications, specifications, progress reports, and final report.

(H) a recognition of the need for, and an ability to engage continuing professional development

The students often must utilize the internet to learn and apply the new technologies that they have chosen in support of their projects

(I) an ability to use the current techniques, skills, and tools necessary for computing practice.

Projects use current computing and modeling/design tools.

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

The students are required to use standard design and development principles on a significant software

Topics covered:

.

- Introduction to Expert Systems
- Knowledge Representation, Introduction to CLIPS Language
- Methods of Inference, Implementation in CLIPS
- First Order Logic, Resolution, Examples of Expert Systems, Implementation in CLIPS
- Introduction to Fuzzy Expert Systems, Fuzzy Logic, Fuzzy Set ad Fuzzy numbers, Fuzzy Inference Systems, Examples of Fuzzy Expert Systems
- Design of Commercial Expert Systems using CLIPS

Credits and contact hours: 3 Credits (2 30-minutes lectures perweek)

Instructor: khalilalsulbi ,alialomari

Textbook :
Robert Sebesta, Programming the World Wide Web, 2011, ISBN-10: 0132130815
Stepp,Miller,Kirst. Web Programming Step by Step.(1st Edition, 2009)

***Specific courseinformation*:**

d. This is a practical course that will enable students to develop skills in website development and administration, exploring backend/server technologies such as (PHP/ASP and XML, JavaScript, CSS and web framework). The course will focus on building dynamic websites and issues relating to user input validation, authorization, roles management, database connectivity and session and state management.

e. *Prerequisite*: 6001216-3 – Multimedia Systems .
f. Required, elective, or selected elective:

***Specific goals for thecourse*:**

c. *Specific outcomes ofinstruction*:
- Students will be able to construct websites that receive and perform complex processing of user input on the server side.
- Students will be appreciate the different methods of storage available for data required and served by web applications.
- Students will be able to create websites with interactivity without page reloading.
- Students will be able to configure a modern web server for deploying large web sites.

d. *ABET Criterion 4 Student Outcomes addressed by thecourse*.

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

Students are required design and implement a software project to meet a specification.

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

Projects are implemented in teams.

(F): An ability to communicate effectively;

The project requires a written final report

(I): An ability to use the current techniques, skills, and tools necessary for computing practice.;

Projects use current computing and modeling/design tools.

Topics covered:

- Introduction and revision for XHTML forms
- Cascading Style Sheets (CSS) and themes
- JavaScript and Document Object Model (DOM)
- Web servers management and Administration
- Web Forms and server side scripting
- Web Development Frameworks
- User input Validation
- Database Connectivity
- XML and AJAX
- Session and state management
- Authentication and Authorization
- Web 2.0 Applications and open source applications

6001312-3: Fundamentals of Database Systems

Credits and contact hours: 3 Credits (3*50 minutes lectures perweek, 0 lab hour)

Instructor: SaadAlbagmi

Textbook: Fundamentals of Database Systems, 5th ed., by Elmasri and Navathe, Pearson International Edition, 2007.

***Specific courseinformation*:**

dd. Fundamentals of database design and data indexing techniques. Data models. Data base design theory. Query languages, their implementation and optimization. Database transaction processing.

ee. *Prerequisite*: 6001222-3 - Systems Analysis and Design

ff. Required, elective, or selected elective: None

***Specific goals for thecourse*:**

u. *Specific outcomes ofinstruction*:

a. The student will understand various different types of data modeling techniques and the supporting theoretical foundation.
b. The student will understand how to use different types of query languages.
c. The student will understand a variety of techniques for designing database schemas, associated index structures, and design and implementation of a database system.
d. The student will understand the notions of concurrency control, recovery, and security.

v. *ABET Criterion 3 Student Outcomes addressed by thecourse*:

(A): An ability to apply knowledge of mathematics, computing, science, and engineering appropriate to the discipline;

Students apply knowledge of computing and design to a project

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

Students are required to design and implement a software project to meet a specification.

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

Projects are implemented in teams.

(F): An ability to communicate effectively

The projects require communications, specifications, progress reports, and final report.

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

Projects use current computing and modeling/design tools.

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

The students are required to use standard design and development principles on a significant software project

Topics covered:

- Relational algebra: relations, tuples, attributes, schemas, relational operators and expressions.
- Functional Dependencies: keys, closures, Armstrong's axioms, canonical cover.
- Normalization: anomalies, lossless decomposition, dependency preservation, BCNF, 3NF.
- SQL queries: types, 3 valued logic, nulls, select, ordering, joins, set operators, aggregate functions, grouping, sub-queries.
- SQL data manipulation: insertion, deletion, and update.
- SQL data definition: schema definition, default, primary key, unique, not null, check, assertions, foreign keys, referential integrity, views.
- Transactions: failures, atomicity, consistency, isolation, durability.
- Entity Relationship Modeling: Entities, relationships, attributes, ER diagrams, relationships, participation, fan and chasm traps, roles, weak entities, mapping to relation schemas, is-a relationships and hierarchies.
- Database Indexing

6001213-3: Logic Design & Analysis

Credits and contact hours: 3 Credits (3 x 50 mins lectures, 0 lab hours)

Instructor:

Textbook: Morris Mano, DIGITAL DESIGN, 4th Edition, Prentice Hall, 2007

***Specific courseinformation*:**

a. Computer arithmetic (how data is manipulated by a computer), digital logic and how it relates to boolean algebra, designing of combinational and sequential circuits

a. *Prerequisite*6001101-3 - Introduction to Computer Science
b. Required, elective, or selected elective: None

***Specific goals for thecourse*:**

a. *Specific outcomes ofinstruction*:
   1. Understanding of basic computer arithmetic (how computer manipulates data)
   2. Understanding of digital logic at the gate and switch level
   3. Understanding of combinational and sequential circuits (designing of simple circuits).

b. *ABET Criterion 3 Student Outcomes addressed by thecourse*:

(A): An ability to apply knowledge of mathematics, computing, science, and engineering appropriate to the discipline;
Students apply knowledge of computing and design to a project

(B): an ability to analyze a problem, and identify and define the computing requirements appropriate to its solution

Students apply knowledge of digital logic to develop circuits. Students apply knowledge of computing to practical computing problems.

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

Students are required to design and implement a software project to meet a specification..

(F): An ability to communicate effectively

The projects require communications, specifications, progress reports, and final report.

(J):an ability to analyze the local and global impact of computing on individuals, organizations and society, including ethical, legal, security and global policy issues

Students are required to apply their knowledge of computing to design a solution to a problem and to document the solution including the tradeoffs involved in their design choices

Topics covered:

- Data representation in computer systems (signed and unsigned arithmetic)
- Addition, subtraction, multiplication and division
- Floating-point arithmetic
- Fundamentals of Boolean algebra and logic gates
- Basic concepts of combinational circuits (adders, subtractors, multiplexors, decoders, encoders, magnitude comparator)
- Basic concepts of sequential circuits (flip flops, counters)