CS301 Industrial Training

Module Code: CS301
Module Title: Industrial Training
Level: 3
Credit points: 1
Module Leader: Prof. Ali El-Bastawissy
Pre-requisite: CS102x

Aims

Each student is required to spend a minimum of six weeks of supervised industrial placement in Egypt or abroad, to apply knowledge acquired in his/her course of study and learn practical work experience.

Learning outcomes

Knowledge

On completion of this module, the successful student will be able to:

- Apply knowledge acquired in the first three years of the student's programme in practical environment. (1)
- Demonstrate an enhanced awareness of industrial and commercial practice and the requirements of a professional workplace. (2)
- Gain knowledge from working with practitioners, and learn practical work experience. (3)

Skills

This module will call for the successful student to:

- Work with people at all levels of the profession. (4)
- Gain practical working skills such as teamwork, conforming to corporate disciplines and work practices, in addition to time management. (5)
- Practice the professional and ethical aspects of information technology. (6)
- Develop one's personality from interaction with real business environment and enhance communication skills. (7)

Syllabus

- A programme of supervised training is provided by a company, approved by the University and with monitoring by academic staff of the University. The training should cover at least:
  - Two stages of the software lifecycle.
  - Hardware design and implementation.
  - Any specific technical skills not previously acquired by the student and are required for successful execution of the student's placement duties.
  - Professional skills such as system or network administration, hardware or software maintenance, etc.

Learning, Teaching and Assessment Strategy

A minimum of 6 weeks of work in an approved training situation
At least one and normally two visits to the training establishment by an academic supervisor
Regular contact, and support as appropriate, through electronic communication.

Assessment

The student is required to produce satisfactory report on the work carried out during the placement, and to receive a letter from the placement company attesting to his/her professional conduct.
The course is graded on a Pass/Fail basis. A letter from the placement company confirming attendance and a satisfactory report will earn the student a pass grade.

Learning materials

Software Requirements
As recommended by the placement company.

Useful Websites
As recommended by the placement company.

Reference Text
As recommended by the placement company.

CS313 Data Storage and Retrieval

Module Code:	**CS313**
Module Title:	**Data Storage and Retrieval**
Level:	**3**
Credit points:	**3**
Module Leader:	**Prof. Ali El-Bastawissy**
Pre-requisite:	**CS215**

Aims

For students to gain an understanding of data structures, index design and retrieval issues, to be able to identify fundamental design trade-offs and to apply their acquired knowledge to real world situations, and to properly understand and handle existing implementations of data repositories (e.g. files, Database, and Big Data) structures, indices and queries.

Learning outcomes

Knowledge

On completion of this module, the successful student will be able to:

- Understand issues related to data storage and retrieval for search engines (1)
- Understand efficient techniques to store structured and unstructured data (2)
- Understand indexing structures, methods and techniques (3)
- Understand the different types of queries and how to evaluate and rank data retrieved from queries (4)

Skills

This module will call for the successful student to:

- Be able to integrate a search engine to an information system (5)
- Be able to critique the use of different storage/index/retrieval structures in applications (6)
- Design compatible storage/retrieval application systems according to specific query types using (C++, JAVA, C#, or another language) (7)

Syllabus

- Indexing on disk and B-Trees
- XML
- Hashing
- Information Retrieval Principles
- Crawling
- Text Processing
- Text Indexing
- Search Engine Optimization

Learning, Teaching and Assessment Strategy

Weekly lectures (3 hours per week) to introduce the basic ideas of the course subjects
Weekly tutorials (1.5 hours per week) to discuss the solution of the homework assignments
Weekly computer laboratory (1.5 hours per week) to use a commercial data warehousing tool to solve practical case studies (Microsoft SQL server Analysis Services will be used to develop OLAP cubes and Microsoft Excel for OLAP reporting, …)

Project: students will work in teams to pursue further studies and hands on data warehousing, large data analysis, business intelligence, and data mining. Each team will prepare the project on a subject approved by the instructor.

Assessment

- Unseen Examination: two exams Composed of few questions and a case study to assess the (L.O. 1 to 6).
- In Class Assessment: class discussion for formative assessment and several case studies to train the students on outcomes 2,3,4,6,7
- Lab Project Assessment: to assess (L.O. 5 to 7)

Assessment Weighting

• Unseen Examination	60%
• Case Studies and assignments	20%
• Lab Project Assessment	20%

Learning materials

Reference Text:

- Search Engines Information Retrieval in Practice. Bruce Croft, Donald Metzler and Trevor Strohman, last edition
- File Structures. Michael J. Folk, Bill Zoellick and Greg Riccardi, last edition

Supplementary Readings:

W3schools.com

CS314 Object-Oriented Software Engineering

Module Code: CS314
Module Title: Object-Oriented Software Engineering
Level: 3
Credit Points: 3
Module Leader: Dr. Emad Nabil
Prerequisites: CS214

Aims

This module is designed to introduce the students to the activities involved in a software development project. The module follows an object-oriented approach, compatible with leading programming languages such as Java. Students will be introduced to the concepts and the techniques of the Unified Modelling Language (UML). Advanced modelling concepts and techniques will be used to build complex models. The module project will help the students learn how to work as a team for developing properly designed and documented software systems.

Learning Outcomes

Knowledge

On completion of this module, the successful student will be able to:

- Illustrate the fundamental concepts of object-oriented analysis and design approach (1)
- Demonstrate basic Unified Modelling Language (UML) Notation (2)
- Critically appraise models for object-oriented system development (3)
- Identify system development design patterns (4)

Skills

This module will call for the successful student to:

- Apply the appropriate software analysis and design methodologies to the process of developing large software systems (5)
- Develop formal specifications from informal requirements of software systems (6)
- Design and produce working models of software programmes using UML(7)
- Use CASE tools: to implement the phases of a development methodology, to test design completeness and correctness, and to produce all required documentation (8)

Syllabus

- Introduction to Software Engineering
- Introduction to Unified Modelling Language (UML) Notation
- Object Oriented Systems Analysis and Design based on
  - Use-case modelling (actors, use cases, use case diagram)
  - Domain modelling (class, relationship, inheritance, generalization)
  - Activity modelling (activity diagram)
  - Behavior modelling (sequence / collaboration diagram)
  - State change modelling (state chart diagram)
- Software development life cycle
- Introduction to Design Patterns for System Development
- Software Testing

Learning Teaching and Assessment Strategies

Weekly lectures to introduce the basic concepts of the course subjects

Weekly tutorials: the students are presented with an actual case and are required to apply the course concepts and methods to implement the learned phases of a system design methodology. The instructor will usually play the role of the customer.

Weekly computer laboratory to use automated tools to implement the phases of the system methodology developed in the assignments.

Team Project: The student will work as a member of project team to apply a complete system development methodology for the case study.

Class presentations as part of the implementation of the team project the student will prepare project documentation, prepare and present a slide presentation on the project and give a life demonstration of its operation.

Assessment

Unseen examinations: 3 hours in final and 1.5 hours in Midterm.

Class Exams: are one exam before Midterm and one before Final exam. The unseen examinations and class exams questions are (to assess LOs 1 to 6).

Assignments and team project (to assess LOs 5 to 8) : The students are expected to do incremental practical assignments within a team project in which they apply the methodology learnt to a case study to assess the skill outcomes mentioned above using the chosen CASE tool. The practical work focuses on the application of system development methodology, not programming or application development. The project should be professionally documented and presented.

Assessment Weighting

• Unseen Examinations	60%
• Coursework	40%

Learning Material

Software Requirements

- CASE tool such as Enterprise Architect or Rational Rose.

Useful Websites and books

- http://www.ipd.uka.de/~tichy/patterns/overview.html
- http://wwwbruegge.in.tum.de/OOSE/
- http://www.slideshare.net/SE9/

Reference Text

- Bernd Bruegge, Allen H. Dutoit "Object-Oriented Software Engineering: Using UML, Patterns and Java", Prentice Hall; 3nd edition, 2010.

CS316 Artificial Intelligence

Module Code:	**CS316**
Module Title:	**Artificial Intelligence**
Level:	**3**
Credit points:	**3**
Module Leader:	**Tarek Makladi**
Pre-requisite:	**CS102x**

Aims

Introduction to Artificial Intelligence is a three-credit undergraduate course emphasizing the building of agents, environments, and systems that can be considered as acting intelligently. In particular, you will learn about the methods and tools that will allow you to build complete systems that can interact intelligently with their environment by learning and reasoning about the world.

Learning outcomes

Knowledge

On completion of this module, the successful student will be able to:

- Demonstrate the key components of the artificial intelligence (AI) field.(1)
- Demonstrate the key aspects of intelligent agents.(2)
- Demonstrate the key aspects of constrain satisfaction.(3)
- Demonstrate the key aspects First Order Logic FOL (4)
- Demonstrate and list the key aspects of planning(5)
- Demonstrate the key aspects of Natural Language (6)

Skills

This module will call for the successful student to:

- Solve problems by applying a suitable search method(7)
- Apply mini max search and alpha-beta pruning in game playing.(8)
- Ability to analyze problem specifications and derive appropriate solution techniques for them.(9)

Syllabus

- Agents and environments Ch 1-2
- Search Ch 3-4
- Game playing Ch 5
- Constraint satisfaction Ch 6
- Logical agents, FOL Ch 7
- First order logic Ch 8
- First order inference Ch 9
- Reasoning with uncertainty Ch 13-14
- Planning Ch 10
- Decision making Ch 16 – 17

- Learning Ch 18- 21
- Natural language Ch 22

Learning, Teaching and Assessment Strategy

Weekly lectures will be used to formally introduce the topics of the syllabus and to achieve the learning outcomes but their full understanding is derived from explanation in the lectures combined with recommended readings.

Weekly laboratory sessions will be used to apply the processor design concepts learned in the lectures in order to gain the skills stated in the learning outcomes. Hardware-design software packages are to design, simulate and test the basic internal modules of a generic processor.

Assessment:

Unseen examinations: The exams will be divided between testing the student knowledge outcomes.(L.O.2, 3, 6, 8, 9)

Lab work*:* Lab work will be assessed on the student's ability to use software, design, build and debug the built systems and meet the deadlines.(L.O. 5, 7, 8, 9)

Weekly assignments are exercises on the topics introduced in the lectures and the students will be asked to hand in their solutions. (L.O.1, 3, 5, 7)

Assessment Weighting

Unseen examinations		**60%**
Final Exam	40%	
Mid Term Exam	20%	
Coursework:		**40%**
Lab work	15%	
Assignments	05%	
Quizzes	10%	
Final Project	10%	

Learning materials

Essential
Required Text: Artificial Intelligence: A Modern Approach Third Edition by Russell & Norvig, 2010.

Recommended Readings
Ben Coppin,Artificial Intelligence *Illuminated*, Jones/ Bartlett Publishers, Sudbury, MA, 2004 ISBN 0-7637-3230-3

CS326 Mobile Computing

Module Code: CS326
Module Title: Mobile Computing
Level: 3
Credit points: 3
Module Leader: Prof. Reda Abdel Wahab
Pre-requisite: CS216

Aims

This course introduces the basic concepts and principles in mobile computing. This includes the major techniques involved, and networks & systems issues for the design and implementation of mobile computing systems and applications. This course also provides an opportunity for students to understand the key components and technologies involved and to gain hands-on experiences in building mobile applications.

Learning outcomes

Knowledge

On completion of this module, the successful student will be able to:

- Describe the basic concepts and principles in mobile computing (1)
- Understand the concept of Wireless LANs, PAN, Mobile Networks, and Sensor Networks. 2)
- Explain the structure and components for Mobile IP and Mobility Management. (3)
- Understand positioning techniques and location-based services and applications (4)

Skills

This module will call for the successful student to:

- Describe the important issues and concerns on security and privacy. (5)
- Design and implement mobile applications. (6)
- Design algorithms for location estimations based on different positioning techniques and platforms (7)

Syllabus

- Basic Principles and Concepts in Mobile Computing
- The Concept of Wireless LAN, PAN, Mobile Networks and Sensor Networks
- Positioning Techniques on Different Networks
- Mobility Management and Mobile IP
- Wireless LAN Management
- Device-level Programming

Learning, Teaching and Assessment Strategy

Weekly lectures: to introduce the theoretical concepts of the course subjects.
Weekly tutorials: to discuss the solution of the weekly homework assignments.
Weekly computer laboratory to develop programmes implementing some operating systems functions.
Team Projects: The student works as a member of project team to apply the concepts learned in the course to build one or more functions of a real operating system.

Class presentations: The student is assigned a specific subject to investigate in depth and make a presentation on it in class.

Assessment

- Unseen examinations: exam questions are divided equally between assessing the student understanding of the concepts outlined in the knowledge outcomes and their problem solving abilities in that subject as outlined in the skills outcomes.
- Lab work: The students are expected to use a suitable programming language to apply the concepts learned in the course. They are also expected to do a group project of sizable programming task to assess their practical skills.

Assessment Weights

• In class assessment	20% (L.O. 1,2)
• Lab Projects	20% (L.O. 2,3,7)
• Unseen Examinations	60% (L.O. 3,4,5,6,)

Learning materials

Essential

Asoke K. Talukder, Roopa R. Yavagal, Mobile Computing Technology, Applications, and

Service Creation, McGraw-Hill Communications Engineering, 2005.

Recommended

Martyn Mallick, Mobile and Wireless Design Essentials, Wiley Publishing, 2003.

J. Schiller, Mobile Communications, 2nd edition, Pearson Education, 2003.

D.P. Agrawal and Q.-A. Zeng, Introduction to Wireless and Mobile Systems, Brooks/Cole, Thomson Learning, 2003.

CS334 Programming Concepts and Compiler Design

Module Code:	**CS334**
Module Title:	**Programming Concepts and Compiler Design**
Level:	**3**
Credit points:	**3**
Module Leader:	**Dr. Soha Safwat**
Pre-requisite:	**CS213**

Aims

This module is a comparative study of abstraction, syntax, semantics, binding times, data and sequence control, run-time resources, translators, and storage of programming languages. Also, it provides the detailed theories, principles and practices of the design of compilers.
Students implement a programming project using selected programming languages, to enhance practical aspects.

Learning outcomes

Knowledge

On completion of this module, the successful student will be able to:

- Illustrate the basic components of a programming language. (1)
- Categorize different programming languages considering abstraction, syntax, semantics, binding times, data and sequence control, run-time resources, translators and storage. (2)
- Demonstrate the internals of the process of compilation.(3)
- Explain in detail the structure and components of compilers and implementation of compiler functions.(4)
- Demonstrate and professionally apply techniques of code generation.(5)
- Critically appraise the operation and performance of a compiler.(6)

Skills

This module will call for the successful student to:

- Differentiate between different programming languages. (7)
- Select the appropriate programming language for a given programming problem. (8)
- Learn any programming language faster and easier. (9)
- Use different programming languages to solve a programming problem. (10)

Syllabus

- Preliminaries Evolution of the Major Programming Languages Describing Syntax and Semantics Names.
- Bindings.
- Type Checking and Scopes Data types Expressions and the Assignment Statement.
- Statement-Level Control Structures and implementing sub programmes.
- Steps of compiler Design
  - Lexical Analyzer.
  - Top-Down Parsing.
  - Semantic Analysis.
  - Code Generation.

Learning, Teaching and Assessment Strategy

Weekly lectures to introduce the basic concepts of the course subjects
Weekly tutorials to discuss the solution of the weekly homework assignments
Team Projects: The student will work as a member of project team to apply the concepts learned in the course to real world problems
Class presentations as part of the implementation of the team project the student will be asked to make a presentation of his work.

Assessment

- Unseen examinations: All exam questions are divided equally between assessing the student understanding of the concepts introduced, as outlined in the knowledge outcomes and his problem solving abilities, as outlined in the skills outcomes. (L.O. 1 to 10)
- Lab work: The students are expected to use a suitable programming language to apply the concepts learned in the course. They are also expected to do a group project of a sizable programming task to assess their practical skills. (L.O. 2,4,5,6)

Assessment Weighting

• Unseen Examinations	60%
• In class assessment	20%
• Lab Projects	20%

Learning materials

Essential

- Concepts in programming languages, by John Mitchell, Cambridge University Press, 2003.
- Compiler Construction-Principles and Practice, by Kenneth C. Louden, PWS Publishing Company, 1997.

Recommended

- Essentials of programming languages, 2nd ed., by Daniel P. Friedman, Mitchell Wand and Christopher T. Haynes, The MIT Press, 2000.
- Programming Language Concepts Paradigms, David A. Watt, Prentice Hall PTR,1994.
- Compiler Design by Reinhard Wilhelm and Dieter Maurer, Addison-Wesley, 1995.
- Advanced Compiler Design and Implementation, Steven S. Muchnick, Morgan Kaufmann Publishers, 1997.
- Optimizing Compilers for Modern Architectures, Randy Allen and Ken Kennedy, Morgan Kaufmann Publishers, 2001.

CS351 Operating Systems Concepts

Module Code:	**CS351**
Module Title:	**Operating Systems Concepts**
Level:	**3**
Credit points:	**3**
Module Leader:	**Dr. Soha Safwat**
Pre-requisite:	**CS213**

Aims

The main objective of this module is to introduce important concepts of modern operating systems including processes, concurrent processes, inter-process communication, synchronization, process scheduling and deadlocks, memory management, swapping, paging, segmentation and virtual memory. Also file systems and its implementation besides the input-output systems and mass storage structure.

Learning outcomes

Knowledge

On completion of this module, the successful student will be able to:

- Demonstrate the structure and functions of an operating system. (1)
- Illustrate the methods of process management, CPU scheduling and process synchronization. (2)
- Characterize what is deadlock and how they are handled. (3)
- Describe memory organization and explain memory management techniques. (4)
- Compare between different operating systems. (5)

Skills

This module will call for the successful student to:

- Expertly use any operating system environment. (6)
- Create any operating system component. (7)
- Solve some of the common operating systems problems such as: deadlock, synchronization…etc. (8)

Syllabus

- Operating-System Structures.
- Process Management.
- CPU Scheduling.
- Process Synchronization.
- Deadlocks.
- Memory Management.
- Virtual Memory.

Learning, Teaching and Assessment Strategy

Weekly lectures: to introduce the theoretical concepts of the course subjects.

Weekly tutorials: to discuss the solution of the weekly homework assignments.

Weekly computer laboratory to develop programmes implementing some operating systems functions.

Team Projects: The student works as a member of project team to apply the concepts learned in the course to build one or more functions of a real operating system.
Class presentations: the student is assigned a specific subject to investigate in depth and make a presentation on it in class.

Assessment

- Unseen examinations: exam questions are divided equally between assessing the student understanding of the concepts outlined in the knowledge outcomes and their problem solving abilities in that subject as outlined in the skills outcomes.
- Lab work: The students are expected to use a suitable programming language to apply the concepts learned in the course. They are also expected to do a group project of sizable programming task to assess their practical skills.

Assessment Weights

• In class assessment	20%	(L.O. 1,2)
• Lab Projects	20%	(L.O. 6,7)
• Unseen Examinations	60%	(L.O. 3,4,5,8)

Learning materials
Essential

- Operating Systems Concepts, 9th ed. Abraham Silberschatz, Peter Bear, Galvin Greg Gagne, John Wiley & Sons, 2012.

Recommended

- Modern Operating Systems, 2nd ed. By Andrew Tanenbaum, Prentice Hall, 2001.
- Operating Systems Internals and Design Principles, 3rd ed. by William Stallings, Prentice Hall, 1998.
- Operating Systems: A Modern Perspective Lab Update, 2nd ed. By Gary Nutt, Adison-Wesley, 2001.

CS352   Advanced Operating System

Module Code: **CS352**
Module Title: **Advanced Operating System**
Level: **3**
Credit points: **3**
Module Leader: **Dr. Soha Safwat**
Pre-requisite: **CS351**

Aims

This course expands the principles of operating systems introduced in the prerequisite to cover the advanced topics in modern operating systems, real time, multimedia systems, networks, distributed operating systems, distributed mutual exclusion, distributed deadlocks detection, load balancing, process migration, file management and organization, security and protection, fault tolerance, issues within client/server processing and object orientation.

Learning outcomes

Knowledge

On completion of this module, the successful student will be able to:

- Demonstrate basic concepts commonly used in network operating systems and network programming. (1)
- Critically appraise the advantages and limitations of peer to peer and server based NOS's. (2)
- Categorize and appraise security and protection techniques. (3)
- Discuss advanced features of OS such as client/server processing, object orientation and fault tolerance. (4)

Skills

This module will call for the successful student to:

- Provide a critical analysis of commercially produced NOSs from the perspective of suitability for various applications. (5)
- Select, implement and manage NOSs. (6)
- Select NOS suitable for a particular application. (7)

Syllabus

- Network Structures.
- Distributed System Structures).
- Distributed System Structures
- Protection
- Security
- Fault Tolerance.
- Client/Server processing and Object Orientation.

Learning, Teaching and Assessment Strategy

Weekly lectures: to introduce the basic concepts of the course subjects.
Weekly tutorials: to discuss the solution of the weekly homework assignments.
Weekly computer laboratory: to develop programmes applying the problems
Class presentations the student will be asked to make a class presentation of one of the modules subjects, to assess the skill outcomes.

Assessment

Unseen examinations: exam questions are divided equally between assessing the student understanding of the concepts of theory of computation as outlined in the knowledge outcomes and their problem solving abilities in that subject as outlined in the skills outcomes.
Tutorial assignments: are used for training and formative assessment of the knowledge outcomes.
Lab work: The students are expected to use a suitable programming language to apply the concepts learned in the course.

Assessment Weighting

• Unseen Examinations		60%	(L.O. 1,3,5,6)
• Coursework		40%	(L.O. 2,3,4,7)
• Lab work	20%		
• Presentation	20%		

Learning materials

Essential

- Operating Systems Concepts, 9th ed. Abraham Silberschatz, Peter Bear, Galvin Greg Gagne, John Wiley & Sons, 2012.

Recommended

- Modern Operating Systems, 2nd ed. By Andrew Tanenbaum, Prentice Hall, 2001.
- Operating Systems Internals and Design Principles, 3rd ed. by William Stallings, Prentice Hall, 1998.
- Operating Systems: A Modern Perspective Lab Update, 2nd ed. By Gary Nutt, Adison-Wesley,

CS364 Cloud Computing

Module Code:	**CS364**
Module Title:	**Cloud Computing**
Level:	**3**
Credit points:	**3**
Module Leader:	**Dr. Soha Safwat**
Pre-requisite:	**CS351**

Aims

This module covers computing in the cloud. Unlike traditional computing, this cloud computing model isn’t PC-centric, it is document-centric. Students will learn about the programming necessary for supporting transactional web applications in the cloud -- mission-critical activities that include customer orders and payments.

Learning outcomes

Knowledge

On completion of this module, the successful student will be able to:

- Evaluate cloud computing technologies (1)

- Determine cloud computing components.(2)
- Assess cloud infrastructure and tools (3)
- Criticize enterprise web application using cloud computing (4)

Skills

This module will call for the successful student to:

- Contrast cloud services (5)
- Develop cloud services (6)
- Select an existing virtualization infrastructure (7)
- Develop N-Tier web application (8)
-

Syllabus

- Overview of Distributed Computing
- Introduction to Cloud Computing
- Infrastructure as a Service (IaaS)
- Platform as a Service (PaaS)
- Software as a Service (SaaS)
- Cloud issues and challenges

Learning, Teaching and Assessment Strategy

Weekly lectures: to introduce the theoretical concepts of the course subjects.

Weekly tutorials: to discuss the solution of the weekly homework assignments.

Weekly computer laboratory to develop programmes implementing some operating systems functions.

Team Projects: The student works as a member of project team to apply the concepts learned in the course to build one or more functions of a real operating system.

Class presentations: The student is assigned a specific subject to investigate in depth and make a presentation on it in class.

Assessment

- Unseen examinations: exam questions are divided equally between assessing the student understanding of the concepts outlined in the knowledge outcomes and their problem solving abilities in that subject as outlined in the skills outcomes.
- Lab work: The students are expected to use a suitable programming language to apply the concepts learned in the course. They are also expected to do a group project of sizable programming task to assess their practical skills.

Assessment Weights

Lab weekly progress to assess (L.O. 3, 4, 5, 6, 8)
Project defence to assess (L.O. 2, 4, 6, 8)

Assessment Weights

Coursework	40%
Unseen Examinations	60%

Learning Material

Reference Text:

Miller, Michael (2008) Cloud Computing: Web-Based Applications That Change the Way You Work and Collaborate Online, 1st edition, Que.

Supplementary Readings:

- Reese, George (2009) Cloud Application Architectures: Building Applications and Infrastructure in the Cloud, 1st edition, O'Reilly Media, Inc.
- Ritting house, John (2009) Cloud Computing, 1st edition, CRC Press

CS381 Computer Graphics

Module Code:	**CS381**
Module Title:	**Computer Graphics**
Level:	**3**
Credit points:	**3**
Module Leader:	**Dr. Ahmed Farouk**
Pre-requisite:	**CS232**

Aims

This module aims to introduce geometric computing field which is applied in real world, in different applications like Archi-CAD, or geometric based animation. The nature of this module combines different sectors like mathematics, data structures, and multimedia.

Learning outcomes

Knowledge

On completion of this module, the successful student will be able to:

- Critically analyse different techniques and algorithms used in computer graphic to represent 2D and 3D objects. (1)
- Appreciate the optimization methods, used in graphic algorithms.(2)
- Understand deeply different mathematic techniques behind geometric computing of graphical objects.(3)

Skills

This module will call for the successful student to demonstrate:

- Ability to develop graphical applications.(4)
- Ability to develop programmes for geometric based applications(5)
- Ability to manipulate 2D and 3D graphics.(6)
- Effective use of debugging methods in graphical applications and their challenging projects.(7)
- Effective verbal presentation of ideas and research skills (8)

Syllabus

- Drawing primitives
    - Line (DDA, and Bersenham algorithms)
    - Circle (Direct, Polar, and Midpoint algorithms)
    - Ellipse(Direct, polar and Midpoint algorithms)
    - Curve (Spline, and Bezier algorithms)
- Filling Methodologies
    - Raster (flood-fill algorithm)
    - Vector (general Filling Algorithm)
- 2D Transformations Methodologies
    - Translation
    - Rotation
    - Scaling
    - Reflecting
    - Shearing

Learning, Teaching and Assessment Strategy

Weekly lectures: to introduce the basic concepts of the course subjects listed in the syllabus part.

Weekly computer laboratory: to apply the concepts learned to develop workable programming solutions for different types of problems from a variety of fields, e.g. mathematical, text manipulation and business problems.

Weekly assignments: The student will be assigned a weekly programming homework to develop on his own. All programs have to be submitted to the instructor running without errors.

Project: The student will work as a member of project team to apply the concepts learned in the course to a real world problem. The subject of the project will be chosen to reflect current issues of software development.

Research: the student will select a topic in computer graphics and make a research document, programming demo, and presentation on it.

Oral Exam: to ensure that all math proofs are well understood by the students.

<u>Assessment criteria:</u>

- Unseen examinations: 4 hours in Final.
- Class Exam: from 1 to 2 exams one before midterm and another after midterm.
- Coursework: distributed between the following topics.
    - Lab work, Research, Oral, Tracing, Homework assignments, Project

Assessment Weighting

• Unseen examinations (L.O. 1, 2, 3)	50%
• Coursework:	
• Lab work (L.O. 1, 2, 3)	5%
• Research (L.O. 8)	10%
• Oral (L.O. 3 , 4)	7.5%
• Tracing (L.O. 3, 4)	7.5%
• Homework assignments. (L.O. 1, 2, 6)	10%
• Project (L.O. 1, 2, 3 , 6 , 7)	10%

Learning materials

Software Requirements

- Microsoft visual studio .NET, or any programming language according to instructor's requirements

Useful Websites

- www.computer.org/cga/

Essential

- Fundamentals of Computer Graphics, 2nd Ed. by Peter Shirley, Michael Ashikhmin, Michael Gleicher, and Stephen Marschner (Hardcover - Jul 2005).

Supplementary readings

- Procedural Elements in Computer Graphics, 2nd ed. by David F. Rogers, McGraw Hill, 1997.
- Mathematics for Computer Graphics by John Vince (Paperback - Sep 30, 2005).
- Computer Graphics: Theory Into Practice by Jeffrey J. McConnell (Hardcover - Aug 18, 2005).
- Edward Angel: Interactive Computer Graphics: A Top-Down Approach Using OpenGL, Fifth edition, Publisher: Addison Wesley, ISBN: 9780321535863.

CS403 Advanced Algorithms

Module Code:	**CS403**
Module Title:	**Advanced Algorithms**
Level:	**4**
Credit points:	**3**
Module Leader:	**Dr. Soha Safwat**
Pre-requisite:	**CS213**

Aims

The main objective of this module is to analyse the asymptotic performance of algorithms, write rigorous correctness proofs for algorithms and apply important algorithmic design paradigms and methods of analysis.

Learning outcomes

Knowledge

On completion of this module, the successful student will be able to:

- Argue the correctness of algorithms using inductive proofs and invariants (1)
- Explain the different ways to analyse randomized algorithms (expected running time, probability of error).(2)
- Describe the divide-and-conquer paradigm and explain when an algorithmic design situation calls for it. (3)
- Describe the dynamic-programming paradigm and explain when an algorithmic design situation calls for it. Recite algorithms that employ this paradigm. Synthesize dynamic-programming algorithms, and analyse them. (4)

Skills

This module will call for the successful student to:

- Analyse worst-case running times of algorithms using asymptotic analysis. (5)
- Analyse randomized algorithms. Employ indicator random variables and linearity of expectation to perform the analyses. Recite analyses of algorithms that employ this method of analysis. (6)
- Compare between different data structures. Pick an appropriate data structure for a design situation (7)

Syllabus

- Complexity
- Automata, computability, and complexity theories.
- Insertion sort, complexity analysis
- Asymptotic behaviour
- Efficiency: formal and mathematical definitions and properties
- Growth functions, bounding summations, recurrences, graphs, trees, etc.
- Dynamic Programming
- Matrix Chain Multiplication Problem
- Amortized Analysis; Aggregate Analysis
- Dynamic Tables

Learning, Teaching and Assessment Strategy

Weekly lectures: to introduce the theoretical concepts of the course subjects.

Weekly tutorials: o discuss the solution of the weekly homework assignments.
Weekly computer laboratory: to develop programmes implementing some operating systems functions.
Team Projects: The student works as a member of project team to apply the concepts learned in the course to build one or more functions of a real operating system.
Class presentations: The student is assigned a specific subject to investigate in depth and make a presentation on it in class.

Assessment

- Unseen examinations: exam questions are divided equally between assessing the student understanding of the concepts outlined in the knowledge outcomes and their problem solving abilities in that subject as outlined in the skills outcomes.
- Lab work: The students are expected to use a suitable programming language to apply the concepts learned in the course. They are also expected to do a group project of sizable programming task to assess their practical skills.

Assessment Weights

•	Unseen Examinations	60% (L.O. 3,4,5,6,7)
•	In class assessment	20% (L.O. 1,2)
•	Lab Projects	20% (L.O. 2,3)

Learning materials

Essential

- Thomas Cormen, Charles Leiserson, Ronald Rivest, and Cliff Stein, Introduction to Algorithms, McGraw Hill Publishing Company and MIT Press, 2009 (3rd Edition).

Recommended

- Michael T. Goodrich and Roberto Tamassia, *Algorithm Design: Foundations, Analysis, and Internet Examples*, John Wiley & Sons, Inc., 2002.
- Gilles Brassard, *Fundamentals of algorithmics*, Englewood, N.J. : Prentice Hall, c1996.
- Robert Sedgewick, *Algorithms*, Reading, Mass : Addison-Wesley, c1983.
- Alfred V. Aho, *Data structures and algorithms*, Reading, Mass, Addison-Wesley, c1983.

CS401 Computer Security

Module Code:	**CS401**
Module Title:	**Computer Security**
Level:	**4**
Credit points:	**3**
Module Leader:	**Tarek Makladi**
Pre-requisite:	**CS351**

Aims

This module addresses the problem of securing computer systems. Different levels of computer threats and different authentication methods are studied. Ciphering and cryptographic techniques are studied to create secure algorithms. In addition, web security is introduced for the student to be aware of the different security techniques used at present.

Learning outcomes
Knowledge

On completion of this module, the successful student will be able to:

- Characterize ciphering and cryptology.(1)
- Illustrate the concepts of Hash Function, Message Digest and Message Authentication Code. (2)
- Discriminate between different authentication methods used for access control in computer systems.(3)
- Discriminate between different layers of security.(4)
- Illustrate the concepts of Internet Firewall. (5)

Skills

This module will call for the successful student to :

- Apply key management techniques.(6)
- Propose, apply and evaluate security, privacy and integrity policies for a system.(7)
- Choose and implement the appropriate ciphering and cryptographic techniques.(8)
- Implement different authentication methods.(9)

Syllabus

- Symmetric Block Ciphers (Ch. 3)
- Hash Function, Message Digest and Message Authentication Code (Ch. 4)
- Asymmetric Public-key Cryptosystems. (Ch. 5)
- Public-key Infrastructure. (Ch. 6)
- Network Layer Security. (Ch. 7)
- Transport Layer Security: SSLv3 and TLSv1. (Ch. 8)
- Electronic Mail Security: PGP, S/MIME. (Ch. 9)
- Internet Firewalls for Trusted Systems. (Ch. 10)

Assessment:

Unseen examinations: The exams will be divided between testing the student knowledge outcomes.(L.O.2, 3, 4, 5, 6, 8)

Lab work: Lab work will be assessed on the student’s ability to use software, design, build and debug the built systems and meet the deadlines.(L.O. 2, 3, 7, 8)

Weekly assignments are exercises on the topics introduced in the lectures and the students will be asked to hand in their solutions. (L.O.1, 4, 5, 7, 9)

Assessment Weighting

- Unseen examinations 60%
    - Final Exam 40%
    - Mid-Term Exam 20%

- Coursework: 40%
    - Lab work 15%
    - Assignments 05%
    - Quizzes 10%
    - Final Project 10%

Learning materials

Essential

Internet Security: Cryptographic Principles, Algorithms and Protocols, by Man Young Rhee, John Wiley & Sons, 2003.

Recommended

- Network and Internet security by Vijay Ahuja, Academic Press Ltd., 2004.
- Cryptography and Network Security ,4th ed , by William Stallings, Prentice Hall, 2007.

Software Requirements

- VC++, Java

Useful Websites

- www.rfc.org
- http://www.alw.nih.gov/Security/security-www.html.

CS405x Graduation Project I

Module Code:	**CS405x**
Module Title:	**Graduation Project I**
Level:	**4**
Credit points:	**3**
Module Leader:	**Prof. Ali El-Bastawissy**
Pre-requisite:	**Senior Standing**

Aims

The Graduation Project is designed to give the student the industry experience of working as part of a group of programmers or computer professionals developing an IT project.

The aim of the graduation project is to allow the student to work individually and with a group to acquire new knowledge independently and apply the knowledge and skills he learned in a real life project such as: systems, prototypes ,embedded systems, network based systems, games, application software, etc.

GP is a two-courses project (CS405 and CS406) taken in two successive semesters, in the first course, the student chooses a project subject, and prepares the project proposal including the detailed objective expected outcome. They also do the literature search and the design work for the project. They should present the project interim report at the end of the semester.

A complete description of the project requirement, procedures, and assessment scheme is detailed in the MSA Graduation Project Handbook.

Learning Outcomes

Knowledge

On completion of this module, the successful student will be able to:

- Identify and select a challenging idea for the project that is related to current state of the art in the computing field (1)
- Independently research the underlying theory and practices relevant to the chosen project (2)

Skills

This module will call for the successful student to:

- Transform real world user and domain requirements into well-defined, doable and manageable project specifications.(3)
- Develop, build and test quality software (4)
- Apply the organization and communication skills required to work as member of a project team such as running meetings, making collective decisions, time and people management, writing reports, and giving presentations.(5)
- Apply the principals and practices of software engineering and project management learned during the student course of study.(6)
- Prepare professional system documentation and technical reports.(7)

Syllabus

There is no specific syllabus for the graduation project modules (CS405, CS406), but in CS405 the student is expected to carry out the following tasks:

- Form a group of 1-3 students (if required)
- Choose a subject, research it and submit a proposal
- Do the preliminary literature survey, analysis and design work and start building the project (if applicable)
- Prepare and submit the interim report at the end of the semester.
- Represent their project ideas and execution methodology.

Learning, Teaching and Assessment Strategy

Students in CS405 (The first step of producing final graduation project) are divided into groups; each group is assigned a supervisor. The students will have regular weekly meetings with their supervisor to present and discuss their progress. The supervisor might give the students few informal orientation lectures to clarify the project implementation procedures, project management practical skills and writing and presentation skills.

The students submit weekly progress reports for comments and approval by their supervisor and the project module leader (usually the dean of the faculty or the chairman of the Department).

Laboratory and library facilities are provided for the students to work independently. They are expected to compile an individual project portfolio for each student work.

At the end of the semester the students submit a group interim report.

Assessment

A detailed assessment scheme is devises for the project modules it is detailed in The MSA Graduation Project Handbook.

If the student completes the requirements of CS405 he receives “I” grade (incomplete) pending the completion of the project.

If the student work is not satisfactory he may be asked to re-register for CT405.

The supervisor records the student’s performance in CS405 in respect to the criteria listed in the Handbook such as continuous progress, independent work, etc. These records are taken into consideration when evaluating the project at the end of CS406.

Assessment Weighting

Project defence and Documentation ……100 %

Learning materials

Essential

- Varied

Recommended

- Object-Oriented Software Engineering: Using UML, Patterns and Java 2nd Ed. by Bernd Bruegge and Allen H. Dutoit, Prentice Hall, Oct 5, 2003.

CS406 Graduation Project II

Module Code:	**CS406**
Module Title:	**Graduation Project II**
Level:	**4**
Credit points:	**4**
Module Leader:	**Prof. Ali El-Bastawissy**
Pre-requisite:	**CS405x**

Aims

The Graduation Project is designed to give the student the industry experience of working as part of a group of programmers or computer professionals developing an IT project.
The aim of the graduation project is to allow the student to work individually and with a group to acquire new knowledge independently and apply the knowledge and skills he learned in a real life project such as: systems, prototypes, embedded systems, network based systems, games, application software, etc.
GP is a two courses project (CS405 and CS406) taken in two successive semesters, in the first course, the student chooses a project subject, and prepares the project proposal including the detailed objective expected outcome. They also do the literature search and the design work for the project. They should present the project interim report at the end of the semester.
A complete description of the project requirement, procedures, and assessment scheme is detailed in the MSA Graduation Project Handbook.

Learning Outcomes

Knowledge

On completion of this module, the successful student will be able to:

- Identify and select a challenging idea for the project that is related to current state of the art to the computing field (1)
- Independently research the underlying theory and practices relevant to the chosen project (2)

Skills

This module will call for the successful student to:

- Transform real world user and domain requirements into well-defined, doable and manageable project specifications.(3)
- Develop, build and test quality software (4)
- Apply the organization and communication skills required to work as member of a project team such as running meetings, making collective decisions, time and people management, writing reports, and giving presentations.(5)
- Apply the principals and practices of software engineering and project management learned during the student course of study.(6)
- Prepare professional system documentation.(7)

Syllabus

There is no specific syllabus for the graduation project modules (CS405, CS406), but in CS405 the student was expected to carry out the following tasks:

- Form a group of 1-3 students(if required)
- Choose a subject, research it and submit a proposal
- Do the preliminary literature search, analysis and design work and start building the project (if

applicable)

- Prepare and submit the interim report at the end of the semester.In CS405x, the project team perform a project execution plan in which time, responsibilities, and activities are determined. So, project execution work is broken into sets of individual tasks.

In CS406, the individual student is then expected to carry out the following tasks:

- Do the in depth study for the individual tasks
- Write, test, run the programmes.
- Test and run the complete project and get the outcomes
- Prepare and submit the technical report concerning the project and the individual achieved tasks.
- Present the project concepts, methodologies and individual achieved tasks.

Learning, Teaching and Assessment Strategy

Students in CS406 (The completion of the final graduation project) have to complete the development and testing of their project then complete the project final report. The students will continue to have regular weekly meetings with their supervisor and submit weekly progress reports. The students may consult the faculty of language technical writing consultation unit on writing styles, etc.

Laboratory and library facilities are provided for the students to work independently. They are expected to compile an individual project portfolio for each student work.

Final Report: At the end of the semester, each individual student submits his final project report for evaluation by the project supervisor and the project module leader. The individual project report includes the interim report of CS405x, and the individual student work achieved during CS406.

Project defense: Each student has to submit his completed project to a defense committee composed of an external examiner and several of the faculty members of the MSA. The defense consists of a formal presentation, a comprehensive demo of the project and discussion. The defense is an open event where any attending staff or students can discuss the project with the project group.

<u>Assessment</u>

A rubric of the detailed assessment scheme is devises for the project modules. It is detailed in the MSA Graduation Project Handbook.

The project has to be completed and works properly. If the project is not working properly, the group/student will be asked to re-register for CS406 to get a working project

The supervisor records the student’s performance in CS406 in respect to the criteria listed in the MSA’s Graduation Project Handbook.

The defence committee will evaluate each student individually according to the assessment scheme. The two assessments will be then combined to structure the student final grade.

It is understood that some assessment criteria is best assessed by the supervisor based on continuous monitoring of the students along the course of executing the project, while others can be assessed by the committee. This is taken into consideration in the calculation of the final grade.

Assessment Weighting

Project Course Work (By supervisor and module leader)	…….. 40 %
Project defence and Documentation (By the committee)	…… 60 %

Learning materials

Essential

- Varied

Recommended

- Object-Oriented Software Engineering: Using UML, Patterns and Java 2nd Ed. by Bernd Bruegge and Allen H. Dutoit, Prentice Hall, Oct 5, 2003.

CS411 Theory of Computing

Module Code:	**CS411**
Module Title:	**Theory of Computing**
Level:	**4**
Credit points:	**3**
Module Leader:	**Dr. Soha Safwat**
Pre-requisite:	**CS334**

Aims

This module is an introduction to Computer Science Theory. Topics covered include the basics of the Automata Theory and the Theory of Grammars, to design language definers, differentiate between different statements and different languages.

Learning outcomes

Knowledge

On completion of this module, the successful student will be able to:

- Demonstrate the elements of Automata Theory (Finite State, Pushdown and Turing machines). (1)
- Characterize the limitation of each automata type. (2)
- Relate the theory of grammars to automata theory. (3)

Skills

This module will call for the successful student to:

- Design different language definers (automata, grammars, regular expressions), as well as transform one into another. (4)
- Evaluate the validity of a given statement in automata theory and prove or disprove them. (5)
- Differentiate between regular, context-free, decidable and undecidable languages. (6)

Syllabus

- Introduction to Languages and Grammars.
- Deterministic and Nondeterministic Finite Automata (DFA and NFA).
- Equivalence between DFA and NFA.
- Introduction to Regular Expressions.
- Equivalence between Regular Expressions and NFA.
- Closure Properties of Regular Languages.
- Pumping Lemma and non-regular Languages.
- Introduction to Context-free Grammars and Languages. Derivations Trees and Parsing.
- Transforming Grammars. Chomsky and Greibach normal Forms.
- Introduction to Nondeterministic Pushdown Automata (NPA). Equivalence between NPA and Context-free languages.
- Introduction to Turing machines. Decidable Languages and Computable Functions. Church Thesis.
- Nondeterministic Turing Machines. Universal Turing Machines.
- Un decidability: The Halting Problem.
- Introduction to Computational Complexity. The O-Notation.

Learning, Teaching and Assessment Strategy

Weekly lectures to introduce the basic concepts of the course subjects.

Weekly tutorials to discuss the solution of the weekly homework assignments.
Weekly computer laboratory to develop programmes applying the problems
Class presentations the student will be asked to make a class presentation of one of the modules subjects, to assess the skill outcomes.

Assessment

Unseen examinations: exam questions are divided equally between assessing the student understanding of the concepts of theory of computation as outlined in the knowledge outcomes and their problem solving abilities in that subject as outlined in the skills outcomes.
Tutorial assignments: are used for training and formative assessment of the knowledge outcomes.
Lab work: The students are expected to use a suitable programming language to apply the concepts learned in the course.

Assessment Weighting

• Unseen Examinations		60%	(L.O. 1,3,5,6)
• Coursework		40%	(L.O. 2,3,4)
• Lab work	20%		
• Presentation	20%		

Learning materials

Essential

- Introduction to the Theory of Computation, Second Edition Michael Sipser, Course Technology, 2006.
- Introduction to Automata Theory, Languages, and Computation, 2nd edition.

CS427 Advanced Graphics and Visualization

Module Code:	**CS427**
Module Title:	**Advanced Graphics and Visualization**
Level:	**4**
Credit points:	**3**
Module Leader:	**Dr. Ahmed Farouk**
Pre-requisite:	**CS381**

Aims

This module covers basic concepts of 3D visualization and modelling techniques for 3D objects. It introduces geometric computing for 3D models which is applied in real world, in different applications like 3D games, or geometric based 3D tools.

Learning outcomes

Knowledge

On completion of this module, the successful student will be able to:

- Compare/contrast the different ways of camera synthesis.(L.O. 1)
- Explain the rendering process of complicated 3D models using primitives. (L.O. 2)
- Create complicated animation behaviours using 3D transformations. (L.O. 3)
- Critically analyse and manipulate 3D models. (L.O. 4)

Skills

This module will call for the successful student to demonstrate:

- Ability to develop and import 3D object models from different tools to the programme.(L.O. 5)
- Ability to View geometric of 3D models. (L.O. 6)
- Professional treatment of geometric modelling, lighting, and shading. (L.O. 7)
- Professional implementation of collision detection and response techniques. .(L.O. 8)
- Effective verbal presentation of ideas and research skills (L.O. 9)

Syllabus

- Vertex Buffer
- Projection techniques (parallel, perspective)
- 3d transformations(scaling, rotation, etc)
- Camera model
- Mesh manipulation
- Illumination techniques and texture mapping.

Learning, Teaching and Assessment Strategy

Weekly lectures to introduce the basic concepts of the course subjects listed in the syllabus part.

Weekly computer laboratory to apply the concepts learned to develop workable programming solutions for different types of problems from a variety of fields, e.g. mathematical, text manipulation and business problems.

Weekly assignments The student will be assigned a weekly programming homework to develop on his own. All programmes have to be submitted to the instructor running without errors.

Project: The student will work as a member of project team to apply the concepts learned in the course to a real world problem. The subject of the project will be chosen to reflect current issues of software development.

Research: the student will select a topic in computer and make a research document, programming demo, and presentation on it.
Oral Quizzes: to ensure that all math proofs are well understood by the students.

Assessment

Unseen examinations: 4 hours in Final.
Quizzes: from 1 to 2 quizzes one before midterm and another after midterm.
Coursework: distributed between the following topics.

Lab work
Research
Oral
Tracing
Homework assignments.
Quizzes
Project

Assessment Weighting

• Unseen examinations	50 %
• Course work:	
• Research. (L.O.. 1, 2, 3,4, 8)	7.5%
• Oral . (L.O.. 1, 2, 3, 4, 6,7, 8)	7.5%
• Tracing. (L.O.. 1, 2, 3, 4)	7.5%
• Homework assignments. . (L.O.. from 1to 7)	7.5%
• Lab work and Project. (L.O.. from 1 to 7)	20%

Learning materials

Software Requirements

- Microsoft visual studio .NET, or any programming language according to instructor's requirements

Useful Websites
www.computer.org/cga/
Reference Text

- 3D Programming for Windows, by Charles Petzold, Microsoft Press (July 25, 2007)
- Game Graphics Programming,1st Ed. by Allen Sherrod (Jun 26, 2008).
- Supplementary readings
- Foundations of 3D Graphics Programming: Using JOGL and Java3D, 2nd Ed .By Jim X. Chen, Chunyang Chen. Springer (August 25, 2008)
- Introduction to 3D Graphics & Animation Using Maya (Graphics Series),1st Ed . by Adam Watkins. Charles River Media(April 6, 2006)
- Fundamentals of Computer Graphics, 2nd Ed. by Peter Shirley, Michael Ashikhmin, Michael Gleicher, and Stephen Marschner (Hardcover - Jul 2005).
- Mathematics for Computer Graphics by John Vince (Paperback - Sep 30, 2005).
- Computer Graphics: Theory Into Practice by Jeffrey J. McConnell (Hardcover - Aug 18, 2005).
- Procedural Elements in Computer Graphics, 2nd ed. by David F. Rogers, McGraw Hill, 1997.

CS475 Data Mining

Module Code: **CS475**
Module Title: **Data Mining**
Level: **4**
Credit points: **3**
Module Leader: **Dr. Ismail H. A. Fattah**
Pre-requisite: **MTH204**

Aims

This module emphasizes the concept of Data mining. It aims to analyse large volumes of data and pick out relevant information for decision making. The student will be able to understand basic data mining concepts and principles, in addition to analyzing large databases using the appropriate software.

Learning outcomes

Knowledge

On completion of this module, the successful student will be able to:

- Demonstrate the common data mining techniques.(1)
- Illustrate the use and expected outcomes of applying data mining to different data sets. (2)
- Demonstrate the theory and algorithms used in data mining models (3)
- Compare/ contrast different inference mechanisms to extract the relevant information to assist in decision-making on the basis of patterns and expectations resulting from the data collected (3)

Skills

This module will call for the successful student to:

- Analyse large volumes of data using such technologies as: Machine learning, Statistics, Pattern Recognition, Artificial Intelligence, and Database Systems. (3)
- Develop appropriate models for data mining (4)

Syllabus

- Decision Tree Construction.
- Association Analysis.
- Clustering.
- Rule Induction.
- Bayesian Methods
- Dealing with Noise and Real-Valued Attributes.
- Data Mining from Very Large Databases.

Learning, Teaching and Assessment Strategy

Weekly lectures (3 hours per week): to introduce the basic ideas of the course subjects.
Weekly tutorials (1.5 hours per week): to discuss the solution of the weekly homework assignments.
Weekly computer laboratory (1.5 hours per week) to apply the concepts learned in the course.

Assessment Weighting

• Unseen Examination (L.O.. 2, 3, 5)	60%
• In Class Assessment (L.O.. 2, 3, 5)	20%

- Lab Project Assessment (L.O.. 1, 4, 5, 6, 7) 20%

Learning materials

Reference Text

- Introduction to Data Mining: Concepts and Techniques, Pang-Ning Tan, Michael Steinbach, and Vipin Kumar, Addison Wesley, 2007.

Supplementary Readings

- Ming-Syan Chen, Jiawei Han, and Philip Yu, Data Mining: An Overview from a Database Perspective, IEEE Transactions on Knowledge and Data Engineering, Volume 8, Number 6, December 1996.
- Usama Fayyad, Gregory Piatetsky-Shapiro, Padhric Smyth, and Ramasamy Uthurusamy, Advances in Knowledge Discovery and Data Mining, AAAI Press, 1996.
- Jiawei Han and Micheline Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann, 2000.
- Tom Mitchell, Machine Learning, McGraw-Hill, 1997.
- Ian Witten and Eibe Frank, Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, Morgan Kaufmann, 2000.
- Xindong Wu, Knowledge Acquisition from Databases, Ablex Publishing Corp., U.S.A., 1995.

CS484 Human Computer Interaction

Module Code: CS484
Module Title: Human Computer Interaction
Level: 4
Credit Points: 3
Module Leader: Dr. Ismail H. A. Fattah
Prerequisites: CS314

Aims

This module introduces the field of human computer interaction with emphasis on its impact on software design. It provides the student with theories and models of the way users think and work to guide the students to best design the interface to suite users' preferences. It provides an understanding of the underlying processes of human perception, information processing, and demonstrates their relevance to user interface design. Students will learn how to apply mechanisms such as feedback, user support, navigation aids and good screen design in constructing interface designs that match users' needs. Students will also learn techniques for evaluating user interface designs that are grounded in theory.

Learning Outcomes

Knowledge

On completion of this module, the successful student will be able to:

- Differentiate between the different scientific fields involved in interaction design. (1)
- Illustrate the principles and the applications of ID design goals, usability goals, user experience etc. (2)
- Analyze how much the theories of how people communicate and work can influence the design of interactive systems. (3)
- Illustrate the different methodologies used in interface design and users involvement.(4)

Skills

This module will call for the successful student to:

- Select models that are appropriate to particular design problems and contexts and justify those choices. (5)
- Apply standard usability evaluation techniques to evaluate and critique designs from a usability perspective, and to propose improvements. (6)
- Design interactive systems that are usable and meet the users' needs. (7)

Syllabus

- What is Interaction Design?
- Understanding and Conceptualizing Interaction
- Cognitive Aspects
- Interfaces and Interactions
- The Process of Interaction Design
- Design, Prototyping and Construction
- Design Evaluation: Usability Testing, Field Studies and Analytical Evaluation
-

Learning Teaching and Assessment Strategy

Weekly lectures to introduce the basic ideas of the course subjects

Weekly Lab & Tutorial: The students are given a series of exercises and case studies to allow them to practice HCI & ID subjects discussed in the lectures. Many of these cases involve evaluating existing applications and/or comparing websites and web applications.

Assessment

Assessment will be based on:

- two unseen exams: each composed of 2-3 questions and 1-2 case studies to assess the student ability to apply the module materials (L.O.. 1 to 7)
- Lab case studies: to assess (L.O. 3 to 7)

Assessment Weighting

• Unseen examinations	60%
• Coursework	40%

Learning materials

Essential

- Interaction Design: Beyond Human-Computer Interaction, 3rd ed. by Helen Sharp, Yvonne Rogers, and Jenny Preece, Wiley June 07, 2011

Recommended

- Human-Computer Interaction 3rd ed. by Alan Dix, Janet E. Finlay, Gregory D. Abowd, and Russell Beale, Prentice Hall, Dec 20, 2003
- Designing Interactions by Bill Moggridge, The MIT Press, Oct 1, 2007
- About Face 3: The Essentials of Interaction Design, 3rd ed. by Alan Cooper, Robert Reimann, and David Cronin, Wiley, May 7, 2007
- Designing for Interaction: Creating Smart Applications and Clever Devices (VOICES) by Dan Saffer Peachpit Press, Jul 28, 2006
- The Design of Everyday Things by Donald A. Norman, Basic Books, Sep 17, 2002.

CS486 Image Processing

Module Code: **CS486**
Module Title: **Image Processing**
Level: **4**
Credit points: **3**
Module Leader: **Dr. Ahmed Farouk**
Pre-requisite: **MTH106**

Aims

This module introduces an understanding of image processing techniques and familiarizes with computer vision applications. Practical experience is acquired in the design and implementation of image processing algorithms.

Learning outcomes

Knowledge

On completion of this module, the successful student will be able to:

- Conceptualize image processing problems. (1)
- Demonstrate the image processing concepts. (2)
- Illustrate the standard image processing algorithms.(3)
- Compare/contrast the different applications of computer vision. (4)

Skills

This module will call for the successful student to:

- Illustrate and apply the concepts and methodologies for the formation, representation, enhancement and analysis of digital images. (5)
- Use available tools to develop applications of image processing. (6)
- Improvise the design and implementation of image processing algorithms to suite specific applications. (7)
- Using benchmark images for algorithm testing.(8)

Syllabus

- Image sensing and acquisition
- Image Sampling and Quantization
- Basic Relationship between Pixels.
- Linear & Non Linear Operations
- Image Enhancement in Spatial Domain
- Histogram Processing
- Spatial Filtering (Smoothing, Sharpening)
- Image Restoration
- Image Compression Models
- Edge detection
- Morphological operations
- Error-Free Compression
- Loose Compression
- Image Segmentation

Learning Teaching and Assessment Strategy

Weekly Lectures: The module consists of class lecture sessions and a laboratory component.

Weekly Lab: The lab component of the module consists of assignments and a small project that the students can do on their own time schedule.

For the project component of the image-processing course, every student is supposed to complete a small project.

The principal objectives of completing a project are:

- Students gain practical experience on how to manipulate images,
- Students develop a sense of how image processing solutions are prototyped in software.

For the project, students should use a programming language or a combination of a programming language and MATLAB.

Assessment

- Unseen Examinations a set of question and problems to assess (L.O. 1 to 6)
- In class assessment formative discussion to assess (L.O. 1 to 6)
- Project is used to assess all outcomes (L.O. 1 to 8) the project will be evaluated bon the quality of its design and implementation, Report and presentation

Assessment Weighting

• Unseen Examinations		60%
• In class assessment		20%
• Project		20%
• Overall Project Quality	10%	
• Report	5%	
• Presentation	5%	

Learning materials

Lecture notes

Essential

- “Digital Image Processing, Third edition”, by R. Gonzalez and R. Woods, 2007, Prentice Hall

Periodicals, Web Sites, etc

- IEEE Trans. on Image Processing
- IEEE Trans. on Medical Imaging
- IEEE Trans. on Pattern Analysis and Machine Intelligence
- IEEE Trans. on Biomedical Engineering
- IEEE Trans. on Information Technology in Biomedicine
- IEEE Trans. on Signal Processing
- IEEE Trans. on Visualization and Computer Graphics
- Signal Processing, Image Communication (Elsevier Science)
- Signal Processing (Elsevier Science)
- Pattern Recognition, (Pergamon-Elsevier)
- Pattern Recognition Letters (Elsevier)
- Digital Image Processing websites.