



MSA University

Faculty of Computer Science

Programme Handbook

Modules Description

2015 – 2016

Modules Index

CS100x Introduction to Information Technology.....	6
CS101x Fundamentals of Computing I	8
CS102x Fundamentals of Computing II	10
CS205 Principles of Information Systems.....	16
CS213 Algorithms and Data Structures	18
CS214 Systems Analysis and Design	20
CS217 Professional Computing Ethics	28
CS232 Multimedia Programming.....	30
CS283 Web Programming	32
CS301 Industrial Training.....	34
CS313 Data Storage and Retrieval	36
CS314 Object-Oriented Software Engineering.....	38
CS316 Artificial Intelligence.....	40
CS326 Mobile Computing.....	42
CS334 Programming Concepts and Compiler Design	44
CS344 Component Based Computing	46
CS347 Software Requirements and Specifications.....	48
CS351 Operating Systems Concepts.....	50
CS352 Advanced Operating System	52
CS353 Systems Programming.....	54
CS361 Signal Processing.....	56
CS362 Knowledge Representation and Reasoning	58
CS364 Cloud Computing.....	61
CS372 Information Integration and Data Warehousing	63
CS373 Geographical Information Systems	65
CS381 Computer Graphics	67
CS382 Web Content Management.....	69
CS384 Advanced Web Programming.....	71
CS385 Web Engineering.....	73
CS391 Modeling and Simulation	75
CS393 Operations Research.....	77
CS401 Computer Security.....	79

CS405x Graduation Project I.....	83
CS406 Graduation Project II.....	85
CS411 Theory of Computing.....	88
CS425 Service-Oriented Computing	92
CS427 Advanced Graphics and Visualization.....	94
CS442 Software Construction Quality	96
CS444 Embedded Systems.....	98
CS455 Big Data and Analytics	101
CS458 Software Implementation	103
CS465 Software Project Management	106
CS466 Decision Support and Intelligent Systems.....	108
CS472 Advanced Database Systems	110
CS475 Data Mining.....	112
CS476 Web Database Application.....	114
CS477 Machine Learning	116
CS484 Human Computer Interaction	118
CS485 Pattern Recognition.....	120
CS486 Image Processing	122
CS488 Robotic Interfacing	124
CS489 Semantic Web Programming	126
ENG101x English for Academic Writing Purposes	128
ENG102x English for Study Skills.....	130
ENG201x English for Research Purposes.....	133
MTH100 Calculus	135
MTH103 Discrete Mathematics	137
MTH106 Linear Algebra	139
MTH204 Probability and Statistics	141
PHY103x Fundamentals of Electronics	143
MGT200 Introductory Management.....	145

Computer Science Programme

MODULE CODE	MODULE TITLE	
Computing		
CS100x	Introduction to Information Technology	
CS101x	Fundamentals of Computing I	
CS102x	Fundamentals of Computing II	
CS203	Computer Organization	
CS205	Principles of Information Systems	
CS213	Algorithms and Data Structures	
CS214	Systems Analysis and Design	
CS215	Fundamentals of Database Systems	
CS216	Computer Networks	
CS217	Professional Computing Ethics	
CS232	Multimedia Programming	
CS283	Web Programming I	
CS301	Industrial Training	
CS313	Data Storage and Retrieval	
CS314	Object-Oriented Software Engineering	
CS316	Artificial intelligence	
CS334	Programming Concepts and Compiler Design	
CS351	Operating Systems Concepts	
CS353	Systems Programming	Elective for all
CS361	Signal Processing	Faculty Elective List 2
CS362	Knowledge Representation & Reasoning	Faculty Elective List 2
CS364	Cloud Computing	
CS372	Information Integration and Data Warehousing	Faculty Elective List 1
CS373	Geographical Information Systems	Faculty Elective List 1
CS384	Advanced Web Programming	Elective for CSP
CS391	Modelling and Simulation	Elective for all
CS393	Operations Research	Elective for all
CS401	Computer Security	
CS405x	Graduation Project I	
CS406	Graduation Project II	
CS423	Parallel and distributed systems	Elective for all
CS444	Embedded Systems	Faculty Elective List 2
CS455	Big Data and Analytics	Faculty Elective List 1
CS477	Machine Learning	Faculty Elective List 2
CS472	Advanced Database Systems	Elective for all
CS475	Data Mining	
CS466	Decision Support and Intelligent systems	Faculty Elective List 1
CS484	Human Computer Interaction	
CS485	Pattern Recognition	Elective for all
CS488	Robotic Interfacing	Elective for all
CS490	New Topics in Data Engineering	Faculty Elective List 1
CS491	New topics in Computer Science	Faculty Elective List 2

Computer Science Pathway (CSP)		
CS326	Mobile Computing	
CS381	Computer Graphics	
CS352	Advanced Operating Systems	
CS403	Advanced Algorithms	
CS486	Image Processing	
CS427	Advanced Graphics and Visualisation	
CS411	Theory of Computing	
Internet Computing Pathway (ICP)		
CS382	Web Content Management System	
CS384	Advanced Web Programming	
CS385	Web Engineering	
CS425	Service-Oriented Computing	
CS465	Software Project Management	
CS476	Web Database Application	
CS489	Semantic Web Programming	
Software Engineering Pathway (SEP)		
CS344	Component-Based Computing	
CS347	Software Requirements and Specifications	
CS384	Advanced Web Programming	
CS425	Service-Oriented Computing	
CS442	Software Construction Quality	
CS458	Software Implementation	
CS465	Software Project Management	
Humanities/Social Sciences		
HS Elect I	H/S Elective 1	
HS Elect II	H/S Elective 2	
MGT200	Introductory Management	
Mathematics		
MTH100	Calculus	
MTH103	Discrete Mathematics	
MTH106	Linear Algebra	
MTH204	Probability and Statistics	
Physics		
PHY103x	Fundamentals of Electronics	
Elective Modules In Humanities and Social Sciences		
FAC101	Essentials of Accounting I	
ECO101	Introduction to Economics I	
ECO102	Introduction to Economics II	
BUS102	Business and Society	
MKT201	Marketing I	

CS100x Introduction to Information Technology

Module Code: CS100x
Module Title: Introduction to Information Technology
Level: 1
Credit points: 3
Module Leader: Maged El-Hakeem
Pre-requisite: None

Aims

This module familiarizes the student with the efficient use of computers, to improve general skills and training on popular computer application packages. The module serves also as an introduction to computer-related terminology and introduces software and hardware basics for a computer science student.

Learning outcomes

Knowledge

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

1. Describe the essential hardware components of the computer and its peripheral devices and how they work. (A3)
2. Understand the different numbering systems. (A1)
3. Define the basics of computer communications and networks and describe their applications. (A2)
4. Demonstrate the use of browsers, search engines and Internet learning resources. (A6)
5. Understand programming concepts (Flow chart and pseudo code) (A4)

Skills

6. Use the basic computer productivity tools and applications (word processing, databases, spread sheets, presentations, and graphics. (B2, C2, C4, D3, D5)
7. Use binary arithmetic and conversions between numbering systems. (B3,D4)
8. Effectively use the Internet resources, organise, retrieve, select, set up, and troubleshoot information on a computer. (B2, B4, C2, D3)
9. Examine careers that involve computers and outline a professional development plan. (D6, D8)

Syllabus

- Introduction and historical background.
- Inside the computer - Details about data storage, encoding systems, binary arithmetic, conversions between numbering systems, analysing a computer system, describing the processor.
- Software – common software concepts, purpose and objectives of an operating system, understanding relationship between computers and programming languages, programming concepts and distinguishing between different platforms.
- Storing and retrieving Information, secondary storage files, sequential and direct access, magnetic disks, magnetic tapes, optical laser disks.
- Input /Output devices, traditional input devices, Source-Data automation, output devices, and terminals.
- Computer Networks, a brief overview of data communications hardware, data highways,

network topologies, local area networks.

- An overview of online services & Productivity software. Understanding graphics software concepts, functions of different types of graphics software, multimedia concepts and applications. A detailed practical coverage is provided in lab.
- Conversion between number systems and arithmetic operations.
- Programming concepts. (Flow charts and pseudocode)

Learning, Teaching and Assessment Strategy

Weekly lectures to introduce the basic ideas of the course subjects

Weekly computer laboratory: to investigate the concepts of computer hardware, software and applications practically: To demonstrate the use of computer networks and to learn the use of the standard Desktop applications.

Group presentations Instead of a project, students are assigned a specific subject to investigate in depth and present it in class.

Assessment

Assessment will be based on:

- Two unseen exams several questions to assess the student knowledge and understanding (1, 2, 3, 5, 7, 9)
- Lab weekly assignments and lab presentation to assess (4, 6, 8)

Assessment Weighting

- Unseen examinations 60%
- Coursework 40%

Learning materials

Essential

- Computing Essentials 2013, Complete Edition, by Timothy J. O'Leary, Linda I. O'Leary, McGraw-Hill. ISBN: 978-0-07-131472-5.

Recommended

- Computers: Tools for an Information Age, 8th ed., by H. L. Capron, J. A. Johnson, Prentice Hall, 2003.
- Using Information Technology (A practical introduction to computers and communications), 7th ed., by Brian Williams, Stacey Sawyer, McGraw-Hill, 2007.
- Digital Design, 5th edition, Morris Mano, Prentice Hall, 2007.
- Explorations in Computer Science: A Guide to Discovery by Mark Meyer, Jones & Bartlett Pub, 2003.
- Essential Introduction to Computers, 5th ed. by Thomas J. Cashman and Gary B. Shelly, Course Technology, 2003.

CS101x Fundamentals of Computing I

Module Code: CS101x
Module Title: Fundamentals of Computing I
Level: 1
Credit points: 3
Module Leader: Dr. Ahmed Farouk
Pre-requisite: None

Aims

This module introduces computer programming techniques, with an emphasis on important programming concepts. It gives the student the fundamentals of logic thinking to analyse and solve simple programming problems.

Learning outcomes

Knowledge

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

1. Demonstrate understanding of the main three phases of programme writing (Input, processing, and output).(A2, A3)

Skills

This module will call for the successful student to demonstrate:

2. Provide a correct solution for a given problem in different complicated levels.(B2, B3, B4, C1, C2, D7)
3. Evaluate two different solutions to a given problem and determine a suitable solution.(B2, B3, C1, C3, D1, D2)
4. Ability to differentiate between syntax and semantics and to understand and explain a given source code.(B2, C2, D1)
5. Ability to develop a good programming technique for readability and traceability.(B6, C1, C5)
6. Effectively use automated and manual debugging tools to correct a given programme.(C1, C2, C3, D2)
7. Ability to validate and trace a given source code.(B6, C1, C5)
8. Ability to critically analyse the logic of a programme either to complete it or correct it.(B3,B4,,B5, C3, D1, D2)

Syllabus

- Basic programme construction (identifiers, statements, functions, comments, and pre-processors).
- Data types in a given programming language.
- Constants and variables declaration.
- Input and output statements.
- Output manipulators.
- Mathematical and logical expressions.
- Decision statements.
- Repetition statements.
- One dimensional array.

- String manipulations.
- Procedural programming using user-defined functions.

Learning, Teaching and Assessment Strategy

Weekly lectures to introduce the basic ideas of the course subjects

Weekly computer laboratory: The students are expected to use C++programming language to solve different types of problems from a variety of fields.

Team Projects

Assessment

Assessment will be based on:

Unseen examinations: all exam questions assess the ability of the student to choose the appropriate programming technique for a problem, demonstrate and apply his programming knowledge in problem solving.(to assess 2.,3,6,7,8)

Coursework: distributed between the following topics

Lab work and team projects:

The lab focuses on assessing the practical skills described earlier. All lab work and projects are assessed according to the students' programming ability and efficiency; speed of development, proper use of language constructs, proper structure of the programme, clarity and annotation of the programmes and the quality of the overall solution developed; ease of use and speed of execution. All lab assignments and projects should run correctly, be documented and presented.(to assess 1 to 8)

Homework assignments.(to assess 1 to 8)

In class assessment.(2.,3,6,7,8)

Assessment Weighting

Unseen examinations:	60%
Coursework:	40%
Lab work	15%
Homework assignments	15%
In class assessment	10%

Learning materials

Software Requirements

Microsoft Visual Studio 6 (VC++), or any other programming language.

Reference text

Timothy B. D'Orazio, "Programming in C++ Lessons and Applications", 1st ed. McGraw-Hill Science/Engineering/Math; 2004

Object-Oriented programming in C++, 4th ed. Robert Lafore, Waite Group Press, 2001.

Supplementary readings

C++ Programming: From Problem Analysis to Programme Design, Fourth Edition by D.S. Malik, Course Technology; Feb 11, 2008

C++ Programming: Programme Design Including Data Structures, Third Edition by D.S. Malik W. Norton & Company, May 2, 2006

C Programming: A Modern Approach, 2nd Ed. by K. N. King (Paperback - April 19, 2008)

C++ Programming: An Object-Oriented Approach, by Course Technology, 2003.

Object Oriented Programming Using C++, 2nd Ed. by B. Chandra, Alpha Science International, Ltd - Feb 2005

The C++ Programming Language 3rd ed. by Bjarne, Stroustrup, Addison- Wesley, 2000.

C++ and Object-oriented Numeric Computing for Scientists and Engineers, by Daoqi Yang, Springer Verlag, 2000

Microsoft Visual C++ .Net: Step by Step: by Julian Templeman, Andy Olsen, John Sharp, Microsoft Press, 2003.

CS102x Fundamentals of Computing II

Module Code: CS102x
Module Title: Fundamentals of Computing II
Level: 1
Credit points: 3
Module Leader: Dr. Ahmed Farouk
Pre-requisite: CS101x

Aims

This module provides students with the fundamentals to analyse and solving advanced programming problems. It also aims to provide the students with the ability to design algorithms and translate them into a programming source code.

Learning outcomes

Skills

This module will call for the successful student to demonstrate:

1. Critically design and analyse a large problem using object-oriented methodologies (OOP).(B2, B3, B7, C1, C6, D4, D7)
2. Provide the main three phases of advanced object-oriented programme writing(input, processing and output).(B3, B5, B6, C3, D5)
3. Provide a correct solution for an advanced problem in different complicated levels.(B2, , B3, C1, C3, D2)
4. Evaluate two different solutions to a large problem, using object-oriented methodologies.(B6, B7, C2, C5, D5, D6)
5. Ability to understand and explain an object-oriented source code.(B1, B2, C2, C6, D4, D8)
6. Ability to develop a good programming technique using object-oriented methodologies.(B3, B7, C1, C6, D7, D8)
7. Effective use of automated and manual debugging tools to correct long and high complexity programmes.(B3, C1, C3, D7)
8. A positive contribution to group working.(D2, D3, D6)

Syllabus

- Two-dimensional arrays.
- Structures and arrays of structures.
- Pointers.
- Classes' usage and declaring.
- Objects as function arguments.
- Operator and function overloading.
- Inheritance.

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 students are expected to use the C++ programming language to solve different types of problems from a variety of fields, the lab focuses on assessing the practical skills described earlier. They are expected to do a group project of sizable programming task (300 to 500 line requiring 50 to 100 programming hours). All lab work and projects will be assessed on the student's programming ability and efficiency; speed of development, proper use of language constructs proper structure of the programme, clarity and annotation of the programmes and the quality of the overall solution developed; ease of use and speed or execution. All lab assignment and projects should correctly run, be documented and presented.

Assessment:

Unseen examinations: all exam questions assess the ability of the student to choose the appropriate programming technique for a problem, demonstrate and apply his programming knowledge in problem solving.(3, 4, 5, 6, 7, 8)

Coursework: distributed between the following topics:

Lab project.(from 1 to 8)

Homework assignments (from 1 to 7)

In class assessment (from 1 to 7)

Assessment Weighting

Unseen examinations. 60%

Coursework:

Lab project 20%

Homework assignments 10%

In class assessment 10%

Learning materials

Software Requirements

VC++

Useful Websites

www.cplusplus.com/doc/tutorial/

Reference text

Timothy B. D'Orazio, "Programming in C++ Lessons and Applications", 1st ed. McGraw-Hill Science/Engineering/Math; 2004

Object-Oriented programming in C++, 4th ed. Robert Lafore, Waite Group Press, 2001.

Supplementary readings

C++ Programming: From Problem Analysis to Programme Design, Fourth Edition by D.S. Malik, Course Technology; Feb 11, 2008

C++ Programming: Programme Design Including Data Structures, Third Edition by D.S. Malik W. W. Norton & Company, May 2, 2006

C Programming: A Modern Approach, 2nd Ed. by K. N. King (Paperback - April 19, 2008)
C++ Programming: An Object-Oriented Approach, by Course Technology, 2003.
Object Oriented Programming Using C++, 2nd Ed. by B. Chandra, Alpha Science International, Ltd - Feb 2005
The C++ Programming Language 3rd ed. by Bjarne, Stroustrup, Addison- Wesley, 2000.
C++ and Object-oriented Numeric Computing for Scientists and Engineers, by Daoqi Yang, Springer Verlag, 2000
Microsoft Visual C++ .Net: Step by Step: by Julian Templeman, Andy Olsen, John Sharp, Microsoft Press, 2003.

CS203 Computer Organization

Module Code: CS203
Module Title: Computer Organization
Level: 2
Credit points: 3
Module Leader: Tarek Makladi
Pre-requisite: CS102x

Aims

This module is concerned with Fundamentals of computer operation, instructions set architecture, assembly language programming, computer organization, pipelining, memory hierarchy, storage and I/O, and trends in computer design.

Learning outcomes

Knowledge

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

1. Illustrate the importance of the Instruction Set Architecture abstraction. (A1)
2. Demonstrate an understanding of the instruction set architecture of a MIPS processor. (A2)
3. Demonstrate an understanding of the Procedures and the Runtime Stack. (A5)
4. Demonstrate an understanding of the organization/operation of integer & floating-point units. (A5)
5. Demonstrate a knowledge of mathematics in CPU performance analysis and in speedup computation.(A4)
6. Demonstrate an understanding of the organization/operation of memory and caches. (A3)

Skills

This module will call for the successful student to:

7. Evaluate the performance of processors and caches. (B2, C2, D5)
8. Design the data path and control of a single-cycle CPU.(B2, D5, D7)
9. Design the data path/control of a pipelined CPU & handle hazards.(B2, D5, D7)

Syllabus

- Instruction Set Architecture.
- MIPS Assembly Language Programming.
- Procedures and the Runtime Stack.
- Interrupts.
- Integer Arithmetic and ALU design.
- Floating-point arithmetic.
- CPU Performance.
- Single-Cycle data path and Control Design.
- Pipelined data path and Control.
- Memory System Design.

Learning, Teaching and Assessment Strategy

Weekly lectures (3 hours per week): to introduce the basic ideas of the course subjects.

Weekly computer laboratory (1.5 hours per week): to use simulation tools to apply the concepts learned in the course.

Team Projects The student will work as a member of project team to apply the concepts learned in the course to analyze, design and programming an application for Intel 8051 microcontroller family.

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 live demonstration of its application.

Assessment:

Unseen examinations: The exams will be divided between testing the student knowledge outcomes.(3, 4, 5, 6, 7, 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. (2, 6, 7)

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

Assessment Weighting

- | | | |
|----------------------------|-----|-----|
| • Unseen examinations | %60 | |
| • Coursework | %40 | |
| • Two in class exams | | %10 |
| • Assignments and lab work | | %10 |
| • Team Project Defense | | %20 |

Learning material

Essential

Computer Organization and Design, The Hardware/Software Interface, 4th Edition, by David A. Patterson and John L. Hennessy, ISBN: 9780123744937, Morgan Kaufmann Publishers.

Software

SPIM MIPS

Assembly language simulator; There are also other MIPS editors in the textbook CD

CS205 Principles of Information Systems

Module Code: CS205
Module Title: Principles of Information Systems
Level: 2
Credit Points: 3
Module Leader: Prof. Ali El-Bastawissy
Prerequisites: CS100x

Aims

This course provides an introduction to basic Information System concepts. It explains the importance of information systems in business environment which supports firms' business operation, managerial decision and strategic advantage. It focus on defining main components of information system, basic hardware and software concepts, telecommunications, business process, information systems development, and the application tools.

Learning Outcomes

Knowledge

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

1. Describe the development of computer utilization in the workplace (A4).
2. Identify the key components of an Information System (A2).
3. Discuss how effective use of the Internet and Information Systems furthers the goals and objectives of a business organization operating in a global economy (A4).
4. Understand existing system and Identify potential problems (A5).
5. Figure out the outlines of possible system enhancement (A5).

Skills

This module will call for the successful student to:

6. Prepare Information System case (analyze case studies; identify the potentials and impact of IT evolving the existing system). (B2, B5, C5, D3, D8)
7. Map between system needs and system Software requirements. (B2, B5, C2, C4, D2, D8)
8. Plan, Schedule, and allocate the resources required for information Systems Projects. (B4, B6, C4, D2, D5, D7)

Syllabus

The world of Information Systems

The mangers, the organization, types of users, and the team

Types of IS

Components of IS

Organizing data and information

Systems development lifecycle

IS Methodologies

Comparisons between different IS Methodologies

Information Systems Management

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

Project: The students will work in teams to plan small IS-projects of different types. Some of the better projects will be presented in class.

Assessment

Assessment will be based on:

Two unseen exams composed of several questions to assess the student knowledge and understanding (1 to 7)

Case studies and assignments are used to assess (4,5,6,7,8) 20%

Assignments and term paper to assess (1,2,5,6,7,8) 20%

Assessment Weighting

Unseen examination	60 %
Coursework (no examination)	40 %
• Project	20%
• Assignments and presentation	20%

Learning materials

Essential:

Information Systems Development - Methodologies, tools, and techniques, David Avison & Guy Fitzgerald, 4th Edition, McGraw Hill, 2006

Recommended Readings:

PROJECT MANAGEMENT IN PRACTICE, Samuel J. Mantel, Scott M. Shafer and Jack R. Meredith, Fourth Edition, Wiley, 2010

CS213 Algorithms and Data Structures

Module Code: CS213
Module Title: Algorithms and Data Structures
Level: 2
Credit points: 3
Module Leader: Dr. Ahmed Farouk
Pre-requisite: CS102x

Aims

This module aims to allow the student to analyse and select the optimized algorithm for different problems. Optimization techniques are classified in two ways, either in terms of speed (complexity), or in terms of memory usage (volatile or secondary memory).

Learning outcomes

Knowledge

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

1. Explain in depth the different algorithms for data structure manipulation.(A4, A5)
2. Demonstrate different sort and search algorithms and the optimum search algorithm.(A5)
3. Illustrate the use of recursion and recursion functions.(A5)
4. Categorize the classes of problems according to complexity theory.(A4)
5. Illustrate the use of different data compression techniques for files.(A4, A5)

Skills

This module will call for the successful student to demonstrate:

6. The ability to analyse and select the best algorithm that suits a problem.(B3, B6, C1, C6, D2, D4, D7)
7. The ability to build a robust computer programme that will not crash for unexpected input.(B2, C1, C3, D2, D7)
8. The ability to expertly debug complicated algorithms and programmes.(B3, C1, D2, D7)
9. The ability to understand and examine predefined algorithms.(B3, B7, C1, C2, D1, D3)
10. A positive contribution to group (team) working.(B7, C4, D1, D3, D6)

Syllabus

- Analyse the efficiency of algorithms.
- Recursion functions (implementation and usage).
- Implement a list class.
- Implement a Stack and queues.
- Design of generic classes.
- Recursion functions.
- Searching and sorting algorithms.
- Trees representation.
- Binary search trees

Learning Teaching and Assessment Strategy

Weekly lectures to introduce the basic concepts of the course subjects

Weekly computer laboratory: to provide Hands-on training on the use of C++ programming language to manipulate different file structures and implement sort and search algorithms. Extended time will be given to the student to work on large programming assignment on his own. Team Projects: The student will work as a project team member to apply the learned concepts on real world problems. The subject of the project will be chosen to reflect current issues of software development

Assessment

Unseen examinations.(1, 2, 3, 4, 6)

Coursework: distributed between the following topics.

Lab work(from 1 to 8)

In class assessment.(L.O. 2, 3, 4, 6)

Homework assignments.(1, 3, 4, 6, 7, 8)

Project(1,7, 8, 9,10)

Assessment Weighting

Unseen examinations	60%
Coursework:	
Lab work	8%
In class assessment	5%
Homework assignments	12%
Project	15%

Learning materials

Software Requirements

VC++, or any other programming language.

Useful Websites

www.cplusplus.com/doc/tutorial/

warrior-101.tripod.com/dstut/dstut.htm

Reference Text

Data Structures and Algorithm Analysis in C++, 4th Ed. by Mark Allen Weiss, Addison Wesley
June, 2013

Supplementary Readings

Data Structures and Algorithms in C++, 3rd ed. Adam Drozdek, Course Technology, 2004.

Introduction to Algorithms, 2nd ed. by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, MIT Press, 2001

CS214 Systems Analysis and Design

Module Code: CS214
Module Title: Systems Analysis and Design
Level: 2
Credit points: 3
Module Leader: Prof. Ali El-Bastawissy
Pre-requisite: CS205

Aims

This course emphasizes the system analysis and design techniques for software project development. It includes: setting IS project goals, developing work plans and methods to achieve those goals, and measuring progress against a project plan. Analyse a business need for information and develop an appropriate strategy to solve the problem and provide the required information service. Prepare and use various information gathering techniques for eliciting user information requirements and system expectations. Construct and interpret a variety of system description documents, including data flow diagrams, entity–relationship diagrams, Structured English, structure charts, use-case diagrams, ... etc. The student will design and prototype a system.

Learning outcomes

Knowledge

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

1. Define and differentiate between the concepts of system life cycle development methodology and system modelling. (A4)
2. Illustrate the basic system modelling perspectives and the related modelling primitives.(A3)
3. Demonstrate the main features of structured system modelling perspectives, and their advantages.(A5)
4. Identify the advantages and limitations of different development methodologies.(A5,A7)

Skills

This module will call for the successful student to:

5. Apply the structured system analysis and design techniques to project development and prepare a set of document for the analysis, design and test phases of a project.(B1, B4, C2, D2)
6. Carry out the tasks of information gathering, cataloguing and documenting.(B2, B7, C4)
7. Apply the concepts of data modelling, Process modelling, and Logic Modelling to a software development project. (B1, B3, C1, C2, D7)
8. Manage a system analysis and design project, with reference to project lifecycle Issues.(B6, B7, C5, D6, D7, D8)

Syllabus

- System Development Environment.
- Initiating and Planning Systems Development Projects.
- Information gathering techniques.
- Determining Systems Requirements.

- Structuring System Requirements: Process Modelling and Data flow diagram
- Structuring System Requirements: Logic Modelling and Entity diagram.

Learning Teaching and Assessment

Weekly lectures: will introduce the concepts and topics of the module syllabus.

Weekly Tutorials: the student will be given short case studies to prepare a model based on what has been discussed in the lectures of the same week.

Team Project: the students form groups of 2-3, choose a real application that they are familiar with, and over the length of the semester (~8 weeks) perform the complete system lifecycle studied to prepare system analysis and system design document.

Short paper: starting on the first week of study, each student will prepare a short paper on one the current methodologies such as Agile, RAD, Prototyping etc. The best papers are presented to the class.

Assessment:

Assessment will be based on:

Weekly tutorial assignment (10%) [3,4,5,6,7]

Team projects (20 %) [2,3,4,5,7,8]

Short paper (10%) [1]

Two unseen exams (midterm exam of one hour and a half hours – 20% , and a final exam of three hours – 40%) which include several questions to assess the student knowledge and understanding [2,6,7]

Assessment Weighting

Unseen examinations	60%	
Coursework	40%	
• Weekly tutorial assignment		10%
• Team projects		20 %
• Short paper		10%

Learning materials

Software Requirements

Oracle Designer

Useful Websites

<http://otn.oracle.com/>

www.smartdraw.com

www.uml.org

www.comp.glam.ac.uk

TextBook

Modern Systems Analysis and Design 7th ed. by Jeffrey A. Hoffer, Joey F. George, Joseph S. Valacich, Prentice Hall College, 2013.

Supplementary Readings

- Systems Analysis and Design for the Global Enterprise Bentley and Whitten 2007 McGraw Hill.
- Systems Analysis and Design Methods 6th ed. by Jeffrey L. Whitten, Lonnie D. Bentley, Kevin C. Dittman, McGraw-Hill, 2008.
- Modern Structured Analysis, by E. Yourdon, 2nd ed. Prentice Hall, 2008.

- Systems Analysis techniques, by B. Robinson & M. Prior, International Thomson Computer Press, 2007.

CS215 Fundamentals of Database Systems

Module Code: CS215
Module Title: Fundamentals Of Database Systems
Module Level: 2
Credit Points: 3
Module Leader: Prof. Ali El-Bastawissy
Prerequisites: CS205

Aims

This module introduces the basic concepts in database system and its architecture. It discusses the different models and different levels of abstractions. Then it introduces the entity-relationship model as a conceptual modelling technique. The main subject of the module is the relational database model, languages and systems.

Learning Outcomes

Knowledge

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

1. Demonstrate the concepts of database management systems.(A3)
2. Explain and appreciate the underlying theory, such as mathematics and logic, relevant to database design, development and evaluation.(A1, A5)
3. Illustrate the relational model using entity relationship diagram (ERD).(A4)
4. Illustrate the elements and syntax of the SQL language and explain their use.(A7)

Skills

This module will call for the successful student to:

5. Model business data using entity relationship diagram (ERD), transform it to the relational model and apply normalization and integrity rules to it. (B1, B2, B5, C4, C5, D1, D7)
6. Use any implementation of the SQL language for data manipulation.(B3, B6, C4, C6, D4)
7. Use an RDBMS (e.g. ORACLE) to implement a relational database schema, a database application, and execute queries.(B4, B7, C3, C4, D6, D8)

Syllabus

- Databases and Database users
- Database system concepts and architecture
- The relational data model, relational constraints, and the relational algebra
- Fundamentals of SQL: DDL and DML
- Data modelling using the entity-relationship model
- The relational database standard ER and EER to relational mapping and other relational languages
- Converting ER/EER models into Relational model
- Fundamentals of Normalization

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 use some RDBMS and its event driven development environment to build database applications, and to practice SQL and ERDs

Team Projects: the project team will apply the concepts learned in the course to build a real database application. The projects will be implemented using ERDs, ORACLE Developer (FORMS & REPORTS), and SQL

Assessment

Assessment will be based on:

Two unseen exams several questions to assess the student knowledge and understanding (1 to 5)

Assignments and Case studies is used to assess (1 to 5) 20%

Team Project to assess (5 to 7) 20%

Assessment Weighting

Unseen examinations 60 %

Coursework 40 %

Assignments and Case studies 20%

Team Project

20%

Learning Materials

Essential

Fundamentals of Database Systems, 6th ed. by Ramez Elmasri and Shamkant B. Navathe, Addison-Wesley, 2011.

Recommended Readings

- An Introduction to Database Systems, 8th ed. by C. J. Date, Addison-Wesley, 2008.
- Database System Concepts by Abraham Silberschatz, Henry F. Korth, and S. Sudarshan McGraw-Hill- Aug 1, 2009.
- Database Systems: Design, Implementation, and Management 8th ed. by Peter Rob and Carlos Coronel, Course Technology, Dec 20, 2007.
- Oracle 10g Developer: PL/SQL Programming by Joan Casteel, Course Technology, Aug 15, 2007.
- Oracle Database 11g: The Complete Reference by Kevin Loney Osborne ORACLE Press Series, Dec 17, 2008.
- Oracle Developer Advanced Forms and Reports, by Peter Koletzke, Paul Dorsey, McGraw-Hill Osborne, 2007.

CS216 Computer Networks

Module Code: CS216

Module Title: Computer Networks

Level: 2

Credit points: 3

Module Leader: Dr. Samir Hassan

Pre-requisite: CS100x

Aims

This module covers the high-level (protocol)oriented aspects of computer networks, specifically: application, transport and network layers. It includes the internet, socket programming and quality of service issues. It also introduces the student to problem analysis and network administration.

Learning outcomes

Knowledge

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

1. Describe communication protocol sand layered network architectures.(A5)
2. Develop networked applications based on client-server and web based techniques.(A7)
3. Administer, and maintain a computer network.(A3)
4. Demonstrate the principals of wired and wireless LAN and WAN.(A3,A4)

Skills

This module will call for the successful student to:

5. Evaluate the techniques Used to transmit data in mobile communication system.(B2, C2, D1)
6. Examine and comprehend the networking concepts including protocol layer stack ,client-server paradigm, application layer applications including Telnet, FTP,DNS,HTTP, SMTP, other state of arts topics including wireless and mobile networks, and computer network administration.(B4, C4, D6)
7. Examine and analyze the transport-layer concepts including transport layer services, reliable. Un-reliable data transfer, TCP protocol and UDP protocol.(B6, C2, D7)
8. Examine and synthesize the network layer concepts including network-layer services, routing, IP protocol and Pad dressing. (B5, C2, D7)
9. Examine and evaluate the data link layer and local area network concepts including data link layer services, Ethernet, error detection and correction and ARP protocol. (B7, C3, D8)

Syllabus

- Computer Networks and the Internet.
 - What is the internet?
 - Protocol layers and their service models.
 - History of computer networking and the internet.

- Application Layer.
 - Principles of Application Layer Protocols.
 - The Web and HTTP.
 - File Transfer :FTP.
 - Electronic Mail in the Internet.
 - DNS-The Internet's Directory Service.
 - Socket Programming with TCP.
 - Socket Programming with UDP.
 - Building a Simple Web Server.
- Transport Layer.
 - Introduction and Transport-Layer Services.
 - Multiplexing and De-multiplexing.
 - Connection less Transport: UDP.
 - Principles of Reliable Data Transfer.
 - Connection- Oriented Transport :TCP.
- Network Layer and Routing.
 - Introduction and Network Service Models.
 - Routing Principles.
 - The Internet Protocol(IP).
 - Routing in the Internet.
 - IPv6.
- Data Link Layer and Local Area Networks.
 - Introduction and Network Service Models.
 - Introduction and services.
 - Error detection and correction techniques.
 - Multiple access protocols.
 - LAN addresses and ARP.
 - Ethernet.
 - Wireless links.
 - PPP :The Point-to-Point Protocol.
- Network Management.
 - What is network management?
 - The infrastructure for network management.
 - The internet network-management framework.

Learning, Teaching and Assessment Strategy

Teaching/learning approaches are integrated with assessment arrangements to facilitate student achievement of the learning outcomes identified for this module.

Lectures (3 hours per week) introduce the key features of Computer networks, relate to relevant current scientific thinking, to open up associated issues, and invite student questions and debate. [1, 2, 3, 5, 7, 8 and 9]

Weekly Tutorials:(1.5 hour per week) to advise and assist student groups in developing group presentation and perform lab exercises. Tutorials will provide the framework to promote student reflection, including production of an individual reflective critique/evaluation of personal learning and of the module. [1, 3, 4 and 7]

Network Lab: In the lab the student will be able to:

1) Investigate the Ethernet protocol and the ARP protocol including capturing and analyzing

Ethernet frames and observing the ARP protocol in action.

2) Explore several aspects of the ICMP protocol including ICMP messages generated by the Ping program, ICMP messages generated by the Trace route program and the format and contents of an ICMP message.

3) Investigate the IP protocol, focusing on the IP datagram.

4) Explore the DHCP and examine the DHCP packets captured by a host.

5) Investigate the behavior of TCP in detail.

6) Take a closer look at the client side of DNS.

7) Explore several aspects of the HTTP protocol including the basic GET/response interaction, HTTP message formats, retrieving large HTML files, retrieving HTML files with embedded objects, and HTTP authentication and security.

8) Implement client-server applications using socket programming.

[3, 6 and 9]

Assessment

1) Assignments that discuss the students' understanding of the classes [3, 5, 6 and 7]: 10%

2) Lab Exercises [6 and 7]: 10%

3) Group Presentation/Team Project to demonstrate the module outcomes through a practical network application [2, 3, 4, 5 and 7]: 20%

4) Unseen examinations: 60%

Learning Materials

Software Requirements:

1) Ethereal Application

Textbook:

- J. Kurose and K. Ross, Computer Networking :A Top- Down Approach Featuring the Internet", Addison-Wesley, 5th edition, Mar 2009.

- Data Communications and Networking, 4th Ed. By Behrouz A. Forouzan, Mc Graw Hill Higher Education, May 1, 2006.

Supplementary Readings:

Data and Computer Communications, 8th ed. By William Stallings, Prentice Hall, 2007.

CS217 Professional Computing Ethics

Module Code: CS217
Module Title: Professional Computing Ethics
Level: 2
Credit Points: 3
Module Leader: Prof. Ali El-Bastawissy
Prerequisites: CS102x

Aims

Information Systems is an area of practical activity, which in different ways employs and affects a large number of people in society. It is now vital that graduating CS students are aware of the most pressing social, legal and professional issues affecting Information Systems. The aim of this module is to provide the students with the tools enabling them to appreciate and incorporate the ethical, legal and professional standards of the computer science profession into their careers.

Learning Outcomes

Knowledge

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

1. Present basic ethical theories governing the field of professional ethics.(A1, A6)
2. Discuss the applicability, benefits and rules of the different intellectual property protection methods available for CS professionals.(A4)
3. Identify the possible improper or criminal behaviours within the computer profession and their consequences.(A2, A6)

Skills

This module will call for the successful student to:

4. Analyse the ethical, legal and professional standing of an activity within the computing field.(B8,C4, D1, D8)
5. Determine the possibility or appropriateness of using an intellectual property protection method such as copyright or patent to protect a given product.(B4, C3, D2, D7)

Syllabus

- Computer ethics and cyber ethics
- Application of normative ethical frameworks to computer issues
- Intellectual property rights including copyright, patent, trade secrets and trade marks
- Regulating and governing the internet
- Free speech and content controls in cyber space
- Computer and Cybercrimes.
- Securing the electronic frontier
- Professional Ethics and Codes of Conduct

Learning Teaching and Assessment Strategy

Weekly lectures to introduce the basic ideas of the course subjects

Ethical presentation: The students are assigned a computer of Internet related issue to investigate and prepare a short paper and presentation of the issue.

Weekly tutorial: The first few weeks will be used to discuss few case studies. The tutorial is mainly used for the above student presentations. The presenting student is evaluated on the quality and completeness of his presentation while attending students are evaluated on their ability to comprehend and criticize the presentation.

Assessment

Assessment will be based on:

- two unseen exams several questions to assess the student knowledge and understanding (1 to 5)
- Case studies is used to assess (1,4) 10%
- Presentation and term paper to assess (3,4,5) 30%
- In addition the presentation and term paper is used to assess the student's general graduate skills such as independent research, technical writing and communication skills.

Assessment Weighting

- | | |
|-------------------------------|------|
| • Unseen examinations | 60 % |
| • Coursework | 40 % |
| • Case studies | 10% |
| • Presentation and term paper | 30% |

Learning materials

Essential

- Cyberethics: Morality And Law in Cyberspace, 5th ed. by Richard Spinello, Jones & Bartlett; Feb 2013

Recommended

- Quinn, Michael J. Ethics for the Information Age, 5th ed. Upper Saddle River, NJ: Pearson, 2013.
- Computer Ethics: A Global Perspective by Giannis Stamatellos, Jones & Bartlett; 1st ed. January 9, 2007
- The Essence of Professional Issues in Computing, by R. Ayres, Prentice Hall, 2007
- A Gift of Fire: Social, Legal, and Ethical Issues for Computing and the Internet 3rd ed. by Sara Baase, Prentice Hall, Jan 6, 2008
- Computer Ethics and Social Value, by D. Johnson and H.F. Nissenbaum, Prentice Hall, 1998.

CS232 Multimedia Programming

Module Code: CS232
Module Title: Multimedia Programming
Level: 2
Credit points: 3
Module Leader: Dr. Ahmed Farouk
Pre-requisite: CS213

Aims

This module investigates and provides an overview of multimedia programming concepts. It aims to two parallel techniques visualization and multimedia techniques. Visualization techniques introduce different methods of programming under GUI (Graphical User Interface) environment. Multimedia techniques introduce skills of animation methods of photometric, and colour images.

Learning outcomes

Knowledge

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

1. Compare/Contrast the main differences between the event handling in software development in relation to console application.(A4)
2. Critically ensure the quality of image appearance and image enhancement.(A4)
3. Critically handle the display snapshots of large sense. (A4)
4. Critically understand different multimedia file format details (for example: waves, bmp, avi) (A5)

Skills

This module will call for the successful student to:

5. Read and write multimedia files with different formats. (B1, C1, D6)
6. Design and implement a large project in small modules using DLL, as well as interactive user interfaces and navigational functions. (B6, B7, C2, D1)
7. Utilize team approaches to problem solving and decision-making. (B4, C2, D3)
8. Plan, design and develop a comprehensive multimedia project. (B6,B7, C5, D8)

Syllabus

- Introduction to GUI programming.
- Event handling.
- Drawing in GUI programming (pens, lines ,fonts and text drawing)
- Double Buffering techniques.
- Using Timers in animation.
- Image photometric.
- Image Transformations.
- Building custom controls.
- Image and Sound file format.
- Playing AVI files.

Learning, Teaching and Assessment Strategy

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

Assessment

- Unseen examinations (2, 3, 4, 5)
- Coursework: distributed between the following topics.
 - Lab work (1, 4, 5)
 - Homework assignments. (1, 2, 3, 4, 5)
 - In class assessment (formative) to assess (1, 2, 3, 4)
 - Project (2, 3, 4, 5 , 6, 7, 8)

Assessment Weighting

- | | |
|-------------------------|-----|
| • Unseen examinations. | 40% |
| • Coursework: | |
| • Lab work. | 20% |
| • Homework assignments. | 10% |
| • Project. | 30% |

Learning materials

Software Requirements

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

Useful Websites

- <http://csharpcomputing.com/>
- <http://www.c-sharpcorner.com/>
- <http://www.programmingtutorials.com/csharp.aspx>

Reference Text

- Programming Microsoft® Windows® Forms, by Charles Petzold, 2005

CS283 Web Programming

Module Code: CS283
Module Title: Web Programming
Level: 2
Credit Points: 3
Module Leader: Dr. Emad Nabil
Prerequisites: CS101x

Aims

This is a core module for web programming. The student will learn the client-side aspect of web programming. The topics will cover different client-based techniques and their applications in real world. Emphasis will be made on Mark up and Scripting languages and their use in web applications.

Learning Outcomes

Skills

This module will call for the successful student to:

1. Evaluate web technologies programming languages used (B1, D1, D6)
2. Implement client-side scripting tools and techniques (B3, D4)
3. Create web applications using the new frameworks such as Asynchronous JavaScript and XML-Ajax (B6, D7)
4. Develop Rich Internet Applications-RIA (B3, D8)
5. Assess the use of Mark-up languages in building web pages (B1, C2)
6. Create interactive web pages using scripting language such as JavaScript (B7, D8)
7. Develop using Document Object Model methods (B3, C6)
8. Evaluate web techniques such as Ajax in development of web applications (B7, C1, C5)

Syllabus

- Extensible Mark-up Languages such as HTML5 and XML
- Cascading Style Sheets (CSS3)
- Client-side scripting languages such as JavaScript
- Server-side scripting language such as ASP.NET
- Document Object Model (DOM)
- Using XML and the DOM
- The use of scripting languages libraries such as J Query
- Creating a full Rich Internet Application (RIA)

Learning, Teaching and Assessment Strategies

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

Weekly computer laboratory: to develop Rich Internet Applications using client-side web programming techniques.

Team Projects: 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 client-side web programming.

Assessment

Unseen Exams: Two unseen exams several questions to assess the student knowledge and understanding (1, 2, 3, 5, 6, 8)

Course work: composed of Assignments, Lab work and team project:

Lab weekly progress to assess (5, 6, 7, 8)

Project defence to assess (3, 4, 6, 7, 8)

Assessment Weighting

- Unseen Examinations 60%
- Coursework 40%

Learning Material

Reference Text

- Deitel, Paul (2012) Internet and World Wide Web: How to Programme, 5th edition, Prentice Hall.

Useful websites

- <http://www.w3schools.com/>
- <http://tutorialspoint.com/>

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:

1. Apply knowledge acquired in the first three years of the student's programme in practical environment. (A1, A4, A6)
2. Demonstrate an enhanced awareness of industrial and commercial practice and the requirements of a professional workplace. (A6)
3. Gain knowledge from working with practitioners, and learn practical work experience. (A2)

Skills

This module will call for the successful student to:

4. Work with people at all levels of the profession. (B1,D3,D6)
5. Gain practical working skills such as teamwork, conforming to corporate disciplines and work practices, in addition to time management. (B3, D1, D8)
6. Practice the professional and ethical aspects of information technology. (C5)
7. Develop one's personality from interaction with real business environment and enhance communication skills. (B7, C2, D4)

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:

1. Understand issues related to data storage and retrieval for search engines (A1, A2)
2. Understand efficient techniques to store structured and unstructured data (A4, A5)
3. Understand indexing structures, methods and techniques (A4, A5)
4. Understand the different types of queries and how to evaluate and rank data retrieved from queries (A4, A5)

Skills

This module will call for the successful student to:

5. Be able to integrate a search engine to an information system (B7, C6, D6)
6. Be able to critique the use of different storage/index/retrieval structures in applications (B6, C3, D2)
7. Design compatible storage/retrieval application systems according to specific query types using (C++, JAVA, C#, or another language) (B4, B7, D5)

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 (1 to 6).
- In Class Assessment: class discussion for formative assessment and several case studies to train the students (2,3,4,6,7)
- Lab Project Assessment: to assess (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:

1. Illustrate the fundamental concepts of object-oriented analysis and design approach (A1)
2. Demonstrate basic Unified Modelling Language (UML) Notation (A3)
3. Identify system development design patterns (A2,A4)

Skills

This module will call for the successful student to:

4. Apply the appropriate software analysis and design methodologies to the process of developing large software systems (B4, B5, C6)
5. Develop formal specifications from informal requirements of software systems (B3, C1, D6)
6. Design and produce working models of software programmes using UML(B7, C5, D8)
7. Use CASE tools: to implement the phases of a development methodology, to test design completeness and correctness, and to produce all required documentation (B2, B4, C5)
8. Critically appraise models for object-oriented system development (B3, C1, D2)

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 live 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 1 to 5).

Assignments and team project (to assess 4 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://www.bruegge.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; 3rd 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:

1. Demonstrate the key components of the artificial intelligence (AI) field.(A1)
2. Demonstrate the key aspects of intelligent agents.(A2)
3. Demonstrate the key aspects of constrain satisfaction.(A2, A7)
4. Demonstrate the key aspects First Order Logic FOL (A4)
5. Demonstrate and list the key aspects of planning(A7)
6. Demonstrate the key aspects of Natural Language (A2)

Skills

This module will call for the successful student to:

7. Solve problems by applying a suitable search method(B2, B4, C1)
8. Apply mini max search and alpha-beta pruning in game playing.(B4, D7)
9. Ability to analyze problem specifications and derive appropriate solution techniques for them.(B3, B4, C1, D3, D4)

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.(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.(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. (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:

1. Describe the basic concepts and principles in mobile computing (A1)
2. Understand the concept of Wireless LANs, PAN, Mobile Networks, and Sensor Networks. (A2, A3)
3. Explain the structure and components for Mobile IP and Mobility Management. (A3, A4)
4. Understand positioning techniques and location-based services and applications (A3, A4)

Skills

This module will call for the successful student to:

5. Apply the important issues and concerns on security and privacy. (B3, C1, C3, D2)
6. Design and implement mobile applications. (B4, B6, B7, C6, D4, D8)
7. Design algorithms for location estimations based on different positioning techniques and platforms (B3, B7, C6, D6, D7)

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% (1,2) |
| • Lab Projects | 20% (2,3,7) |
| • Unseen Examinations | 60% (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:

1. Illustrate the basic components of a programming language. (A1)
2. Categorize different programming languages considering abstraction, syntax, semantics, binding times, data and sequence control, run-time resources, translators and storage. (A3, A4)
3. Demonstrate the internals of the process of compilation.(A5)
4. Explain in detail the structure and components of compilers and implementation of compiler functions.(A5, A7)
5. Demonstrate and professionally apply techniques of code generation.(A7)

Skills

This module will call for the successful student to:

6. Differentiate between different programming languages. (B3, C2)
7. Select the appropriate programming language for a given programming problem. (B7, D6)
8. Learn any programming language faster and easier. (B2, C6, D3)
9. Use different programming languages to solve a programming problem. (B3,C2, C3)
10. Critically appraise the operation and performance of a compiler.(B4, D2)

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. (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. (2,4,5,10)

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. Loudon, 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.

CS344 Component Based Computing

Module Code: CS344
Module Title: Component Based Computing
Level: 3
Credit points: 3
Module Leader: Dr. Ali Fakhry
Pre-requisite: CS314

Aims

The purpose of this module is to teach the advanced principles of design and implementation of component-based systems, based on contemporary methods of development, which is component software, software architectures and middleware platforms.

Learning outcomes

Knowledge

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

1. Identify fundamental concepts, principles and techniques in software reuse.(A1)
2. Explain the value of application programming interfaces (APIs) in software development and apply recognized principles to the building of high-quality software components. (A2)
3. Decide and select architecture for a component-based system suitable for a given scenario.(A5, A6)
4. Identify the kind of event handling implemented in one or more given APIs. (A7)

Skills

This module will call for the successful student to:

5. Use class browsers and related tools during the development of applications using APIs. (B4, C1, D7)
6. Design, implement, test, and debug programmes that use large-scale API packages.(B5, C5, D6)
7. Explain the role of objects in middleware systems and the relationship with components.(B7, C2)
8. Apply component-oriented approaches to the design of a range of software including those required for concurrency and transactions, reliable communication services, database interaction including services for remote query and database management, secure communication and access.(B3, C6, D7, D8)

Syllabus

- APIs
 - Programming using APIs
 - Design of APIs
 - Debugging in the API environment
- Fundamentals
 - The definition and nature of components
 - Components and interfaces
 - Interfaces as contracts
 - The benefits of components
 - Basic techniques
 - Component design and assembly

- Relationship with the client-server model and with patterns
- Use of objects and object lifecycle services
- Use of object brokers
- Marshalling
- Applications
- Patterns as used in analysis and design; context of use including enterprise architectures
- Architecture of component-based systems
- Component-oriented design
- Application frameworks
- Event handling: detection, notification, and response
- Middleware
 - The object-oriented paradigm within middleware
 - Object request brokers
 - Transaction processing monitors
 - Workflow systems
 - State-of-the-art tools

Learning, Teaching and Assessment Strategy

Weekly lectures (3 hours per week): to introduce the basic concepts of the course subjects listed in the syllabus part.

Weekly tutorials (1.5 hours per week): The student will be assigned a weekly programming homework to develop on his own. All assignments have to be submitted to the instructor.

Weekly computer laboratory (1.5 hours per week): to apply the concepts learned to develop workable programming solutions for different types of problems from a variety of fields.

Teams Project: The students are expected to use a programming language to solve different types of problems from a variety of fields; the lab focuses on assessing the practical skills described earlier. They are expected to do a group project of sizable programming task.

Assessment

- Unseen Examinations
- Coursework
 - In Class exams to assess (1,2, 4,6 & 8)
 - Project & defence to assess (1,2, 3, 4, 5, 6, 7 & 8)

Assessment Weighting:

- | | |
|-----------------------|-----|
| • Unseen Examinations | %60 |
| • Coursework | %40 |
| • In Class exams | %20 |
| • Project & defence | %20 |

Learning materials

- Andy JuAn Wang, Kai Qian, “Component-Oriented Programming”, John Wiley and Sons Inc., 2008.
- Ivica Crnkovic, Magnus Larsson, “Building Reliable Component-Based Software Systems”, Arctect House Computing Library, 2007.
- Michael Nash, “Java Frameworks and Components: Accelerate Your Web Application Development”, Cambridge University Press, 2006.
- George T. Heineman, William T. Councill, “Component Based Software Engineering: Putting the Pieces Together”, Addison-Wesley, 2008

CS347 Software Requirements and Specifications

Module Code: CS347
Module Title: Software Requirements and Specifications
Level: 3
Credit points: 3
Module Leader: Prof. Ali El-Bastawissy
Pre-requisite: CS214

Aims

This module focuses on the development of formal and informal specifications for defining software system requirements. Software Requirements Specification (SRS) is a complete description of the behaviour of the system to be developed. Students can examine issues of specification assessment such as consistency, completeness, correctness, and functionality. Several professional issues; customer rights and best practices are also introduced.

Learning outcomes

Knowledge

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

1. Understand formal techniques and metrics to software requirement definition.(A1, A2)
2. Identify the proactive role of the analyst applying best practices and process assessment and improvement to system requirements.(A5)
3. Explain the use of modern object oriented requirement description techniques such as UML, use cases and other diagrams, in addition to mapping system requirements using prototyping.(A7)

Skills

This module will call for the successful student to:

4. Professionally Investigate the art and science of gathering, refining, implementing, and tracking software requirements.(B3, C3, D1, D3)
5. Investigate prototyping to represent a system requirement and gain user approval.(B5, C2, D6)
6. Design requirements using use case diagrams.(B5, B6, C6, D7)
7. Verify and validate requirements definition correctness and completeness, in addition to managing requirements change.(B4, C1, C4, D8)

Syllabus

- Software Requirements Specifications (SRS), Business and User Requirements
- Requirements Development: Elicitation, Analysis, Specification, And Verification
- Prototyping
- New Approaches to Capturing and Describing Requirements and Specifications, Based on the Relationship Between the Software System and the Problem Context.
- The Technology of Description in Software, Including New Ideas Such As Designations, the Separation of Descriptive Moods and the Scope and Span of Description.
- Managing Requirements Change
- Types of Users; Product Champions, Rights and Responsibilities for Software Customers
- Best Practices
- Process Assessment and Improvement

- Object-Oriented Software Development
- Use Cases And Other Diagrams;
- The Unified Modelling Language (UML) advanced features
- Evaluating and Using Requirements Tools; Requirements Traceability Matrix; Impact Analysis

Learning Teaching and Assessment Strategy

Weekly lectures (3 hours per week):to introduce the basic concepts of the course subjects.

Weekly tutorials(1.5 hours per week):to solve problems related to the weekly lecture and explain problems.

Weekly computer laboratory (1.5 hours per week): to use automated tools to represent the requirements definition developed in the assignment. The student will also be required to build at least one prototype for one of the assignment. Both conventional techniques and UML will be used in these exercises.

Team Projects The student will work as a member of project team to apply the complete process of requirement definition for a real business case study. Usually the use of Object Oriented model and the UML will be enforced.

Assessment

- Unseen Examinations
- Coursework
 - In Class exams (1,2, 4, 5 & 7)
 - One Lab work assignment (3 & 6)
 - Project defence to assess (1,2, 3, 4, 5, 6 & 7)

Assessment Weighting

- | | |
|---------------------------|-----|
| • Unseen Examinations | %60 |
| • Coursework | %40 |
| • In Class exams | %15 |
| • One Lab work assignment | %5 |
| • Project defence | %20 |

Learning materials

Essential

- Software Requirement Patterns (Best Practices) by Stephen Withall (Jun 13, 2007)
- Telling Stories: A Short Path to Writing Better Software Requirements by Ben Rinzler , 2009
- Software And Systems Requirements Engineering: In Practice by Brian Berenbach, Daniel Paulish, Juergen Kazmeier, and Arnold Rudorfer, 2009.

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:

1. Demonstrate the structure and functions of an operating system. (A2)
2. Illustrate the methods of process management, CPU scheduling and process synchronization. (A3)
3. Characterize what is deadlock and how they are handled. (A5)
4. Describe memory organization and explain memory management techniques. (A3, A5)
5. Compare between different operating systems. (A3)

Skills

This module will call for the successful student to:

6. Expertly use any operating system environment. (B1, B3, B7, C2, D4)
7. Create any operating system component. (B3, C5, C6, D6, D8)
8. Solve some of the common operating systems problems such as: deadlock, synchronization...etc. (B4, C3, D4)

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% (1,2) |
| • Lab Projects | 20% (6,7) |
| • Unseen Examinations | 60% (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, Addison-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:

1. Demonstrate basic concepts commonly used in network operating systems and network programming. (A2)
2. Appraise the advantages and limitations of peer to peer and server based NOS's. (A5)
3. Categorize and appraise security and protection techniques. (A3)
4. Discuss advanced features of OS such as client/server processing, object orientation and fault tolerance. (A5)

Skills

This module will call for the successful student to:

5. Provide a critical analysis of commercially produced NOSs from the perspective of suitability for various applications. (B1,B4, C5, D4)
6. Select, implement and manage NOSs. (B2, C2, C3, D6)
7. Select NOS suitable for a particular application. (B4, B7, C6, D8)

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% (1,3,5,6)
- Coursework 40% (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, Addison-Wesley, 2001.

CS353 Systems Programming

Module Code: CS353
Module Title: Systems Programming
Level: 3
Credit points: 3
Module Leader: Dr. Ahmed Farouk
Pre-requisite: CS102x

Aims

The main objective is to teach students important concepts of concurrency programming. The benefit of applying this technique to save large computation time, and on the other side the complexity of designing such programming solution through the coding phase and maintains phase.

Learning outcomes

Knowledge

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

1. Differentiate between global and local memory concepts. (A2)
2. Illustrate the role of Communication to do parallel tasks in distributed systems. (A3)
3. Efficiently use of existence number of computers connected by a network to solve problems concurrently. (A3, A7)

Skills

This module will call for the successful student to:

4. Expertly move data from one local memory to another.(B3, C2, D4, D5)
5. Apply Message Passing interface (MPI) to communicate different computers.(B7, C6)
6. Test, monitor and analysis of parallel programme execution.(B3, C1, D4)

Syllabus

- Parallel computing concepts.
- Profilers and performance analysis.
- Object Serialization.
- Concurrency through the thread mechanism.
- Case studies for parallel 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 practical case studies

Assessment

- Unseen examinations: 3 hours in final and 1.5 hours in Midterm Exam.
- Class Exam: One quiz before Midterm Exam and one quiz before final exam All quizzes and exams questions is 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
- 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

- | | |
|---|-----|
| • Practical Lab Assignments (1,3,4,5,6) | 15% |
| • Large Scale Projects (1,3,4,5,6) | 30% |
| • Research work (2) | 15% |
| • Unseen Final Exam (2,4) | 40% |

Learning materials

Essential

- Principles of Concurrent and Distributed Programming, Second Edition (2006), By M. Ben-Ari, Publisher: Addison-Wesley.
- Carlos A. Varela, Gul Agha (2013), Programming Distributed Computing Systems: A Foundational Approach , MIT press

CS361 Signal Processing

Module Code:	CS361
Module Title:	Signal Processing
Level:	3
Credit points:	3
Module Leader:	Dr. Soha Safwat
Pre-requisite:	MTH103

Aims

The module introduces students to mathematical descriptions of signals and systems, and mathematical tools for analysing and designing systems that can operate on signals to achieve a desired effect.

Learning outcomes

Knowledge

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

1. Determine the time domain characteristics of signals and systems (continuous and discrete).(A2)
2. Coverage of continuous and discrete-time signals and systems, their properties and representations and methods those are necessary for the analysis of continuous and discrete-time signals and systems. (A4)
3. Knowledge of frequency-domain representation and analysis concepts using Fourier Analysis tools, Z-transform (A4)

Skills

This module will call for the successful student to:

4. Apply the frequency domain characteristics of signals and systems (continuous and discrete) using Fourier series, Fourier transforms, and Laplace transforms(B3, B7, C2, D4)
5. Perform convolution in both continuous and discrete time.(B3, B7, C3, D6)
6. Demonstrate the ability to generate, analyse, and simulate signals and systems.(B6, C5, C6, D7)

Syllabus

- Signals and Systems Introduction
- Representing Signals Mathematically
- Basic Continuous and Discrete-Time System Properties
- Linear Time-Invariant Systems
- Convolution
- Discrete Fourier transform (DFT)
- Structure of digital filters
- Z transform

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% (1,3,5,6) |
| • Coursework | 40% (2,3,4) |
| • Lab work | 20% |
| • Presentation | 20% |

Learning materials

Essential

- Alan V Oppenheim, Alan S Wilsky and Hamid Nawab S, "Signals & Systems", Prentice Hall
- Haykin and Barry Van Veen, "Signals & Systems", John Wiley and Sons Inc., New Delhi, 2008.

Reference Books:

- Ashok Ambardar, "Introduction to Analog and Digital Signal Processing", PWS Publishing Company, Newyork, 2002.
- John .G.Proakis , "Digital Signal Processing Principles, Algorithms and Applications, Prentice Hall, New Delhi 2006.

CS362 Knowledge Representation and Reasoning

Module Code: CS362
Module Title: Knowledge Representation and Reasoning
Level: 3
Credit points: 3
Module Leader: Dr. Soha Safwat
Pre-requisite: CS316

Aims

The main objective of this module is to acquire a good understanding of the logical foundations of Knowledge Representation and Reasoning as well as to become familiar with current research trends in the field also represent knowledge in artificial intelligence, and the associated methods of automated reasoning.

Learning outcomes

Knowledge

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

1. Understand the fundamental principles of logic-based Knowledge (A1)
2. Understand the notion of a reasoning service (A2)
3. Understand the fundamental trade-off between representation power and computational properties of a logic-based representation language.(A3)
4. Understand how the theoretical material covered in the course is currently being applied in practice.(A7)

Skills

This module will call for the successful student to:

5. Be able to model simple application domains in a logic-based language.(B3, C2, D4)
6. Master the fundamentals of the reasoning algorithms underlying current systems.(B4, C2, C6, D5)
7. Be conversant with several widely used knowledge representation languages(B7, C6, D7)

Syllabus

- Introduction to knowledge-based technologies and knowledge representation.
- Propositional Logic as a simple knowledge representation language
- Reasoning in Propositional Logic.
- Reasoning in Description Logics.
- Rule-based Knowledge Representation and Reasoning.
- Ontologies and Ontology Languages.
- Semantic Web
- Modeling uncertainty
- Classical vs. non-monotonic logic. Ways to achieve non-monotonicity

- Stable Model Semantics
- Probabilistic Inferences

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% (1,2) |
| • Lab Projects | 20% (2,3,7) |
| • Unseen Examinations | 60% (3,4,5,6) |

Learning materials

Essential:

- Knowledge Representation and Reasoning. Ronald Brachman and Hector Levesque. The Morgan Kaufmann Series in Artificial Intelligence, 2004.
- The Description Logic Handbook: Theory, Implementation and Applications, 2nd Edition. Franz Baader, Diego Calvanese, Deborah L. MacGuinness, and Daniele Nardi. Cambridge University Press. 2007

Recommended:

- Foundations of Semantic Web Technologies. Chapman & Hall/ CRC Textbooks in Computing. Pascal Hitzler, Markus Kroetsch, and Sebastian Rudolph, 2009.
- Bridges Between Classical and Nonmonotonic Logic. David Makinson. Texts in Computing. 2005.

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:

1. Understand cloud computing technologies (A1, A2)
2. Determine cloud computing components.(A3)
3. Assess cloud infrastructure and tools (A7)
4. Understand enterprise web application using cloud computing (A5, A6)

Skills

This module will call for the successful student to:

5. Contrast cloud services (B3, C1)
6. Develop cloud services (B4, C2, C3, D1)
7. Select an existing virtualization infrastructure (B5, C6, D2)
8. Develop N-Tier web application (B6, C5, D5, D6)

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 (3, 4, 5, 6, 8)

Project defence to assess (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

CS372 Information Integration and Data Warehousing

Module Code: CS372
Module Title: Information Integration and Data Warehousing
Level: 3
Credit points: 3
Module Leader: Prof. Ali El-Bastawissy
Pre-requisite: CS215

Aims

Data Warehousing has become an essential business tool for making informed decisions. This course introduces students to data warehousing concepts and emphasizes a hands-on approach to reinforce the theory. A large project is used to design and develop a data warehouse. Star schema, fact tables and dimension tables will be examined. Multi-dimensional databases are emphasised. A team project will be used to handle the process of moving data from OLTP systems to a DW with management reports through the cube and pivot tables.

Learning outcomes

Knowledge

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

1. Demonstrate the structural components of a data warehouse. (A2, A3)
2. Demonstrate the techniques for the development and construction of a data warehouse. (A4)
3. Classify the technical design issues involved in data warehouse construction. (A6)
4. Characterize the operating and managerial issues involved in maintaining a data warehouse. (A5)
5. Demonstrate the components of enterprise systems such as on-line analytic processing (OLAP) and data mining systems and contribution of data warehouse to their implementation. (A7)

Skills

This module will call for the successful student to:

6. Design and construct data warehouses. (B1, B4, B5, C5, C6, D8)
7. Use a commercial data warehousing tool to extract and analyse strategic information. (B7, C2, D1, D2, D3)
8. Pursue DW Design and implementation projects. (C2, C4, D3, D6, D8)

Syllabus

- Reasons for data warehousing
- Terminology and Architecture
- Materialized and Virtual data Integration
- Components of a data warehouse
- The data warehouse lifecycle
- DW database conceptual and logical design.
- Data modelling options
- OLAP Cubes Design
- Extraction, Transformation, and Loading (ETL) processes

- Managing the data warehouse

Learning, Teaching and Assessment Strategy

Weekly lectures (3 hours per week) to introduce the basic ideas of the course subjects

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 (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 (6 to 8)

Assessment Weighting

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

Learning materials

Reference Text:

- Data Warehousing in the Age of Big Data, Krish Krishnan, June 2013, The Morgan Kaufmann Series.
- The Data Warehouse Lifecycle Toolkit: Expert Methods for Designing, Developing and Deploying Data Warehouses by Kimball, Ralph et al, Wiley,2008, ISBN 0-471-25547-5

Supplementary Readings:

- Building the Data Warehouse by W. H. Inmon (Paperback - Oct 7, 2005)
- Data Warehousing Fundamentals: A Comprehensive Guide for IT Professionals by Paulraj Ponniah (Hardcover - Aug 3, 2001) ISBN 0471412546 or 9780471412540. There is an e-book version of this text, ISBN 0471221627
- Article by Susan Gallas comparing Kimball and Inmon approaches for DW:
<http://www.dmreview.com>
- The OLAP Report: www.olapreport.com
- Microsoft Online resource: <http://msdn.microsoft.com/library/>
- SQL Server Tutorials www.microsoft.com
- Microsoft SQL Server 2005 Analysis Services Step by Step by Reed Jacobson published by Microsoft Press; ISBN 0-7356-2199-3
- DM Review: www.dmreview.com
- Building the Data Warehouse, W.H. Inmon, John Wiley & Sons, ISBN 0-764-59944-5

CS373 Geographical Information Systems

Module Code: CS373
Module Title: Geographical Information Systems (GIS)
Level: 3
Credit Points: 3
Module Leader: Prof. Ali El-Bastawissy
Prerequisites: CS215

Aims

This module aims to introduce the principles, applications, trends, and pertinent issues of geographic information systems and sciences, including remote sensing (RS), cartography, geography, and global positioning systems (GPS). Students will also learn how to assess the creation and usage of web GIS services namely the Open Geospatial Consortium – OGC. They will understand how to model spatial database and to link it to other applications such as mobile web mapping applications.

Learning Outcomes

Knowledge

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

1. Describe different representations of images: vectors, raster and projections (A1)
2. Implement Spatial database management systems (A2)
3. Create OGC web services (A5)
4. Implement Web GIS, GPS and Mobile web (A5, A7)

Skills

This module will call for the successful student to:

5. Assess maps from the freely available geodata (B3, C2, D6)
6. Create solutions with spatial analysis (B5, C5, D7)
7. Criticize creation and / or usage of OGC web services (B3, B5, C2, D2)
8. Assess connecting GIS spatial database with real time wireless mobile GIS applications (B6, B7, C6, D4)

Syllabus

- Introduction to GIS concepts, tools and techniques
- Maps and Maps analysis
- Vectors, raster and projections
- Geodata and database
- Spatially enabling the database
- GIS Application: Network analysis, 3D GIS and Web GIS
- Creating of OGC web service
- Using of OGC web service
- Keyhole Markup Language (KML) and Geography Markup Language (GML)
- Web maps for mobile such as Google maps

Learning Teaching and Assessment Strategies

Weekly lectures to introduce the basic concepts of the course subjects
Weekly computer laboratory to develop GIS applications.

Team Projects The student will work as a member of project team to apply the concepts learned in the course to a real world problem.

Assessment

Two unseen exams several questions to assess the student knowledge and understanding (1, 2, 3, 4, 6, 7)

Assignments, Lab work and team project:

Lab weekly progress to assess (1, 4, 5, 7, 8)

Project defence to assess (3, 6, 7, 8)

Assessment Weighting

- Coursework 40%
- Unseen Examinations 60%

Learning Material

Reference Text

- Ian Heywood , Sarah Cornelius, Steve Carver; An Introduction to Geographical Information Systems (3rd Edition), 2006

Supplementary Readings

- Davis, Scott (2007) GIS for Web Developers: Adding 'Where' to Your Web Applications, 1st edition, O'Reilly
- Wernecke, Josie (2008) The KML Handbook: Geographic Visualization for the Web, 1st edition, Addison Wesley
- Udell, Sterling (2008) Beginning Google Maps Mashups with Maplets, KML, and GeoRSS: From Novice to Professional, 1st edition, Apress.
- Heywood, Ian (2006) An Introduction to Geographical Information Systems, 3rd edition, Prentice Hall.
- Geographic Information Analysis by David O'Sullivan, David Unwin Mitchell, 2010.
- Tyler (2005) Web Mapping Illustrated: Using Open Source GIS Toolkits, 1st edition, O'Reilly
- Hjelm, Johan (2002) Creating Location Services for the Wireless Web: Professional Developer's Guide, 1st edition, John Wiley & Sons.

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:

1. Understand different techniques and algorithms used in computer graphic to represent 2D and 3D objects. (A1, A2)
2. Appreciate the optimization methods, used in graphic algorithms.(A3)
3. Understand deeply different mathematic techniques behind geometric computing of graphical objects.(A1, A7)

Skills

This module will call for the successful student to demonstrate:

4. Ability to develop graphical applications.(B1, C2)
5. Ability to develop programmes for geometric based applications(B1, B2, C5, D8)
6. Ability to manipulate 2D and 3D graphics.(B2, C6, D5, D6)
7. Effective use of debugging methods in graphical applications and their challenging projects.(B5, B6, C3, D2)
8. Effective verbal presentation of ideas and research skills (B7, C6, D1)

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.

Assessment criteria:

- 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 (1, 2, 3) | 50% |
| • Coursework: | |
| • Lab work (1, 2, 3) | 5% |
| • Research (8) | 10% |
| • Oral (3 , 4) | 7.5% |
| • Tracing (3, 4) | 7.5% |
| • Homework assignments. (1, 2, 6) | 10% |
| • Project (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.

CS382 Web Content Management

Module Code: CS382
Module Title: Web Content Management
Level: 3
Credit Points: 3
Module Leader: Prof. Ali El-Bastawissy
Prerequisites: CS215

Aims

This module aims to provide the student with an understanding of the web content management system (WCMS or Web CMS). Students will learn how to create such a system to manage and control a large, dynamic collection of Web material. They will also learn how to develop modules to incorporate them in the designated system.

Learning Outcomes

Knowledge

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

1. Describe different web content management systems (A1)
2. Understand the best practice to develop modules (A2, A3)
3. Create a web content management system (A5)

Skills

This module will call for the successful student to:

4. Contrast different types of web content management systems based upon objectives, tools and techniques (B3, B7, C1, C6)
5. Assess implementing new modules for existing CMS (B5, B6, C6, D5, D6)
6. Contrast different techniques to develop CMS (B2, C6, D4)
7. Evaluate the use of SOA in developing CMS (B3, C3, D4)
8. Implement SOA in creating CMS (B7, C5, C6, D1, D3)

Syllabus

- Web content management systems – open source and others
- Content types and significance
- Modules development techniques
- Development and implementation of themes
- CMS tools and techniques
- Client-side, server-side and SOA for the CMS
- Publishing the CMS

Learning, Teaching and Assessment Strategies

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

Weekly computer laboratory: to develop a content management system.

Team Projects 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 web content management system.

Assessment

Two unseen exams: several questions to assess the student knowledge and understanding (1, 2, 5, 6, 7)

Course work: composed of Assignments, Lab work and team project:

Lab weekly progress to assess (3, 6, 7, 8)

Project defence to assess (3, 4, 5, 6, 7, 8)

Assessment weighting

- Coursework 40%
- Unseen Examinations 60%

Learning Material

Essential

- Margot Bloomstein, (2012) Content Strategy at Work: Real-world Stories to Strengthen Every Interactive Project

Supplementary Readings

- Brampton, Martin (2008) PHP 5 CMS Framework Development: Expert insight and practical guidance to creating an efficient, flexible, and robust framework for a PHP 5-based content management system, 1st edition, Apress
- Lyman, Forrest (2009) Pro Zend Framework CMS: Building a Full CMS Using Advanced Aspects of the Zend Framework, 1st edition, Apress
- Padilla, Armando (2009) Beginning Zend Framework, 1st edition, Apress
- Zend (2009) Zend Framework: The Official Programmer's Reference Guide, 1st edition, Apress
- Nickull, Duane (2009) Web 2.0 Architectures: What entrepreneurs and information architects need to know, 1st edition, O'Reilly
- Shuen, Amy (2008) Web 2.0: A Strategy Guide: Business thinking and strategies behind successful Web 2.0 implementations, 1st edition, O'Reilly

CS384 Advanced Web Programming

Module Code: CS384
Module Title: Advanced Web Programming
Level: 3
Credit Points: 3
Module Leader: Dr. Emad Nabil
Prerequisites: CS283

Aims

This module serves as an alternative/professional approach of web programming. The student will learn the server-side aspects and web multimedia programming. The topics will cover different server-based techniques and their applications in real world. Emphasis will be made on object-oriented programming and the use of classes in web applications.

Learning Outcomes

Knowledge

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

1. Understand web programming languages used on the server side (A2)
2. Implement server-side programming tools and techniques (A4)
3. Understand the use of class libraries (A1)

Skills

This module will call for the successful student to:

4. Assess server-side technologies in building web pages (B2)
5. Develop interactive web pages using server-side techniques (B5, B7, C4)
6. Select classes and apply design patterns (B3, C2, D6)
7. Evaluate the server-side techniques such as PHP, Ajax and XML,ASP.NET, ADO.NET to develop web applications (B3, B6, C5)
8. Create web applications using client-side and server-side techniques(B7, C6, D1, D3)

Syllabus

- Client frameworks such as Ajax and Server-side programming languages
- Integrating server-side technologies with client-side techniques
- Interactive web pages
- Web multimedia standards using SMIL
- Web multimedia technologies such as Flash and Silverlight
- The use of design patterns
- Semantic Web technologies
- RDF (Resource Description Framework)
- Creating an integrated web Application

Learning Teaching and Assessment Strategies

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

Weekly computer laboratory to develop server-side web programming techniques.

Team Projects 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 server-side web programming.

Assessment

Unseen exams: Two unseen exams several questions to assess the student knowledge and understanding (2, 3, 4, 6)

Course work: Composed of Assignments, Lab work and team project:

Lab weekly progress to assess (1, 3, 5, 6, 8)

Project defence to assess (5, 6, 7, 8)

Assessment Weighting:

- Unseen Examinations 60%
- Coursework 40%

Learning Material

Reference Text

- Deitel, Paul (2012) Internet and World Wide Web: How to Programme, 5th edition, Prentice Hall
- Hebler, John et al (2009) Semantic Web Programming, 1st edition, Wiley

Supplementary Readings

- Bulterman, Dick (2008) SMIL 3.0: Interactive Multimedia for Web, Mobile Devices and Daisy Talking Books, 2nd edition, Springer
- Lerdorf, Rasmus et al (2006) Programming PHP, 2nd edition, O'Reilly Media
- McArthur, Kevin (2008) Pro PHP: Patterns, Frameworks, Testing & More, 1st edition, Deven N. Shah (2009) A Complete Guide To Internet and Web Programming. Dreamtech Press
- Zandstra, Paul (2008) PHP Objects, Patterns, and Practice, 2nd edition, APRESS
- APRESS

Useful websites

- <http://www.w3schools.com/>
- <http://tutorialspoint.com/>

CS385 Web Engineering

Module Code: CS385
Module Title: Web Engineering
Level: 3
Credit Points: 3
Module Leader: Dr. Moustafa M. Elazhary
Prerequisites: CS341

Aims

This module aims to provide the student with an understanding of an agile and adaptable approach to the development of next generation Web Apps-systems that are more complex, more functional, and more significant than any that exists today. It discusses a pragmatic process for engineering web-based systems and applications. It covers the technical methods that will lead to high quality Web Apps produced in a minimum of time and the tools needed to implement a web engineering process within the organization.

Learning Outcomes

Knowledge

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

1. Understand the web engineering process and the web engineering best practices (A1)
2. Create the planning activities (A2)
3. Ability to implement the construction and deployment activities (A5, A7)
4. Ability TO develop using different types of techniques and tools including application frameworks (A5)

Skills

This module will call for the successful student to:

5. Analyse the best practice for a company's web presence (B4, C2, D1)
6. Design and develop interactive websites using application framework tools and techniques (B5, B6, B7, C5, D3)
7. Assess the web application (C2, D2)
8. Test and criticize the performance of a developed website on the Internet and the need to have a scalable solution (B1, C1, D6)

Syllabus

- Requirements engineering for web applications
- Modelling web applications
- Web application architectures
- Technologies for web applications
- The web application development process
- Testing web applications
- Customization and development of themes for application framework
- Development of new modules for application framework

Learning Teaching and Assessment Strategies

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

Weekly computer laboratory to design, develop and modify websites using application framework tools and techniques.

Team Projects 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 engineering web applications.

Assessment

Two unseen exams several questions to assess the student knowledge and understanding (1, 2, 3, 4, 5)

Assignments, Lab work and team project:

Lab weekly progress to assess (6, 7, 8)

Project defence to assess (3, 6, 7, 8)

Assessment Weighting

- | | |
|-----------------------|-----|
| • Unseen Examinations | 60% |
| • Coursework | 40% |

Learning Material

Reference Text

- Pressman, Roger S et al, (2008) Web Engineering: A Practitioner Approach, 1st edition, McGraw-Hill Higher Education.

Supplementary Readings:

- Kappel , Gerti et al, (2006) Web Engineering: The Discipline of Systematic Development of
- Web Applications, 1st edition , John Wiley & Sons.

CS391 Modeling and Simulation

Module Code: CS391
Module Title: Modeling and Simulation
Level: 3
Credit points: 3
Module Leader: Dr. Ossama Hassan
Pre-requisite: MTH204

Aims

This module aims to provide understanding of the concepts of simulating and modelling real systems with emphasis on discrete event simulation. Students will learn to develop solutions for simple simulation problems of real systems.

Learning outcomes

Knowledge

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

1. Ability to analyse system performance using simulation. (A1)
2. Demonstrate an understanding of the concept of random-numbers. (A2)
3. Discriminate between the various techniques for generating random variations. (A3, A5)

Skills

This module will call for the successful student to:

4. Simulate queuing and inventory systems. (B2, B4, C2)
5. Develop simulation software for simple queuing and inventory problems. (B1, B3, C6, D2, D3, D5)

Syllabus

- Simulation software.
- Simulation of queuing systems.
- Simulation of inventory systems.
- Other examples of simulation.
- Random-number generation.
- Random variant generation.

Learning Teaching and Assessment Strategy

Weekly lectures will be used to formally introduce the topics of the syllabus and to achieve the learning outcomes.

Weekly laboratory sessions will be used by students to develop required software projects in order to practice and gain the skills stated in the learning outcomes.

Assessment

Unseen examinations: exam questions are chosen to test the student knowledge outcomes.

Lab work: The software development laboratory exercises are assessed based on the student programming ability and efficiency, meeting the deadlines, annotation of the programmes and the quality of the overall system.

Weekly assignments are exercises on the topics introduced in the lectures and the students will be asked to hand in their solutions.

Assessment Weighting

- Unseen examinations 60%
- Coursework 40%

Learning materials

Essential

- J. Banks, J. S. Carson, B. L. Nelson, and D. M. Nicol, "Discrete-Event System Simulation", 4th edition, Prentice Hall, 2004.

Recommended Readings

- Averill M. Law, David W. Kelton, "Simulation Modelling and Analysis", 3rd edition, McGraw-Hill, 1999.

CS393 Operations Research

Module Code: CS393
Module Title: Operation Research
Level: 3
Credit points: 3
Module Leader: Dr. Magda Daghestany
Pre-requisite: MTH106

Aims

This course will introduce students to deterministic and stochastic models in operations research. Student will learn to formulate, analyse, and solve mathematical models that represent real-world problems.

Learning outcomes

Knowledge

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

1. Ability to formulate a real-world problem as a mathematical programming model (A1)
2. Understand the theoretical workings of the simplex method for linear programming and perform iterations of it by hand. (A2)
3. Understand the relationship between a linear programme and its dual, including strong duality and complementary slackness (A4)
4. Perform sensitivity analysis to determine the direction and magnitude of change of a model's optimal solution as the data change (A7)

Skills

This module will call for the successful student to:

5. Solve specialized linear programming problems like the transportation and assignment problems. (B2, D2)
6. Solve network models like the shortest path, minimum spanning tree, and maximum flow problems. (B6, C6, D7)

Syllabus

- Introduction to Operations Research
- Introduction to Foundation mathematics and statistics
- Linear Programming (LP), LP and allocation of resources, LP definition, Linearity requirement
- Linear Programming – Simplex Method for Maximizing
- Simplex maximizing example for similar limitations, Mixed limitations
- Sensitivity Analysis: Changes in Objective Function, Changes in RHS, The Transportation Model
- Transportation Problems
- Assignment Problems
- Directed graphs and shortest paths

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% (1,2) |
| • Lab Projects | 20% (2,3) |
| • Unseen Examinations | 60% (3,4,5,6) |

Learning materials

Essential

Operations Research: Applications and Algorithms ; Wayne L Winston ; Indian University, 4th edition, 2004.

CS401 Computer Security

Module Code: CS401
Module Title: Computer Security
Level: 4
Credit points: 3
Module Leader: Dr. Samir Hasan
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:

1. Characterize ciphering and cryptology.(A1)
2. Illustrate the concepts of Hash Function, Message Digest and Message Authentication Code. (A1, A2)
3. Discriminate between different authentication methods used for access control in computer systems.(A3)
4. Discriminate between different layers of security.(A7)
5. Illustrate the concepts of Internet Firewall. (A5)

Skills

This module will call for the successful student to :

6. Apply key management techniques.(B2, C2, C3, D4)
7. Propose, apply and evaluate security, privacy and integrity policies for a system.(B3, C2, D6)
8. Choose and implement the appropriate ciphering and cryptographic techniques.(B4, C1, D6, D7)
9. Implement different authentication methods.(B6, C6, D8)

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.(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.(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. (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>.

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:

1. Argue the correctness of algorithms using inductive proofs and invariants (A1, A2)
2. Explain the different ways to analyse randomized algorithms (expected running time, probability of error).(A2)
3. Describe the divide-and-conquer paradigm and explain when an algorithmic design situation calls for it. (A3)
4. 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. (A4)

Skills

This module will call for the successful student to:

5. Analyse worst-case running times of algorithms using asymptotic analysis. (B2, B3, C1, C2, D2)
6. 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. (B3, B4, C2, D2)
7. Compare between different data structures. Pick an appropriate data structure for a design situation (B4, C1, C3, D4)

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: 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

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

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:

1. Identify and select a challenging idea for the project that is related to current state of the art in the computing field (A2, A4, A7)
2. Independently research the underlying theory and practices relevant to the chosen project (A3, A4, A6, A7)

Skills

This module will call for the successful student to:

3. Transform real world user and domain requirements into well-defined, doable and manageable project specifications.(B2, B3, B4, D1, D8)
4. Develop, build and test quality software (B6, B7, C3, D3, D8)
5. 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.(B1, B6, C5, C6, D5)
6. Apply the principals and practices of software engineering and project management learned during the student course of study.(B3, C5, D6, D7)
7. Prepare professional system documentation and technical reports.(B5, C4, D1)

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 devised 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 Documentation100 %

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:

1. Identify and select a challenging idea for the project that is related to current state of the art to the computing field (A2, A4, A7)
2. Independently research the underlying theory and practices relevant to the chosen project (A3, A4, A6, A7)

Skills

This module will call for the successful student to:

3. Transform real world user and domain requirements into well-defined, doable and manageable project specifications.(B2, B3, B4, D1, D8)
4. Develop, build and test quality software (B6, B7, C3, D3, D8)
5. 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.(B1, B6, C5, C6, D5)
6. Apply the principals and practices of software engineering and project management learned during the student course of study.(B3, C5, D6, D7)
7. Prepare professional system documentation.(B5, C4, D1)

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.

Assessment

A rubric of the detailed assessment scheme is devised 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:

1. Demonstrate the elements of Automata Theory (Finite State, Pushdown and Turing machines). (A1, A2)
2. Characterize the limitation of each automata type. (A2)
3. Relate the theory of grammars to automata theory. (A4, A5)

Skills

This module will call for the successful student to:

4. Design different language definers (automata, grammars, regular expressions), as well as transform one into another. (B2, B3, C2, C6, D4)
5. Evaluate the validity of a given statement in automata theory and prove or disprove them. (B4, C3, D2)
6. Differentiate between regular, context-free, decidable and undecidable languages. (B6, C5, C6, D6)

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; introduce the basic concepts of the course subjects.

Weekly tutorials; 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% | (1,3,5,6) |
| • Coursework | 40% | (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.

CS423 Parallel and Distributed Systems

Module Code: CS423
Module Title: Parallel and Distributed Systems
Level: 4
Credit points: 3
Module Leader: Dr Soha Safwat
Pre-requisite: CS352

Aims

This course covers general introductory concepts in the design and implementation of parallel and distributed systems, covering all the major branches such as Cloud Computing, Grid Computing, Cluster Computing, Supercomputing, and Many-core Computing.

Learning outcomes

Knowledge

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

1. be able to design and analyse concurrent algorithms for a variety of problems and computational models (A2)
2. be familiar with the fundamentals of the architecture and systems software of parallel and distributed systems. (A2, A4)
3. have experience with the implementation of parallel applications on several platforms, and be able to evaluate their performance.(A7)
4. Secure computation in the cloud (A4, A6)

Skills

This module will call for the successful student to:

5. Analyse basic distributed algorithm. (B2, B3, C1, C3, D7)
6. design and analyse parallel and distributed applications.(B4, B7, C6, D4, D5)

Syllabus

- Introduction to Distributed Systems
- Distributed System Models and Enabling Technologies
- Memory System Parallelism for Data –Intensive and Data-Driven Applications
- System Architectures
- Parallel Programming
- Map Reduce
- Ordering of events in distributed systems: clock synchronization, logical clocks

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% (1,2) |
| • Lab Projects | 20% (2,3) |
| • Unseen Examinations | 60% (3,4,5,6) |

Learning material

G. Coulouris, J. Dollimore and T. Kindberg, Distributed Systems: Concepts and Design, 4th Edition, Addison Wesley, 2005.

CS425 Service-Oriented Computing

Module Code: CS425
Module Title: Service-Oriented Computing
Level: 4
Credit Points: 3
Module Leader: Dr. Moustafa M. Elazhary
Prerequisites: CS384

Aims

This module aims to provide the student with an understanding of the service-oriented architecture and definition of conceptual services and service blueprints as well as SOA methodology and lifecycles. Students will assemble application components into a network of services to create flexible, dynamic business processes and agile applications across organizations and computing platforms.

Learning Outcomes

Knowledge

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

1. Understand service oriented architecture and service delivery lifecycle (A1, A2)
2. Contrast service-oriented technology concepts (A3)
3. Identify service-oriented architectural model (A5)
4. Implement service-oriented technology languages (A5)

Skills

This module will call for the successful student to:

5. Contrast component-based architecture with web services (B3, C6, D4)
6. Assess the unique dynamics that constitute service-oriented solution logic (B5, C3, D7)
7. Design your services (B1, B3, C1, D5, D6)
8. Develop service-oriented application using programming languages such as Java and .Net (B1, B7, C1, D8)

Syllabus

- Fundamental SOA & service-oriented computing
- Service delivery lifecycle
- Basic WSDL and SOAP concepts plus UDDI, Discovery and Service registries
- Basic REST Service Concepts and Patterns
- The service-oriented architectural model
- The service-orientation design paradigm and related principles
- Fundamental language elements for XML Schema and SOAP
- Creating service-oriented web application

Learning, Teaching and Assessment Strategies

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

Weekly computer laboratory: to develop a service-oriented application.

Team Projects 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 service-oriented computing.

Assessment

Two unseen exams several questions to assess the student knowledge and understanding (1, 2, 3, 5, 7)

Assignments, Lab work and team project:

Lab weekly progress to assess (2, 4, 6, 7, 8)

Project defence to assess (3, 5, 7, 8)

Assessment Pattern

- Coursework 40%
- Unseen Examinations 60%

Learning Material

Reference Text

- SOA: Principles of Service Design, Erl, Thomas (2004) Service-Oriented Architecture: A Field Guide to Integrating XML and Web Services, 1st edition, Prentice Hall.

Supplementary Readings

- Erl, Thomas (2007) SOA Principles of Service Design, 1st edition, Prentice Hall.
- Erl, Thomas (2009) SOA Design Patterns, 1st edition, Prentice Hall.
- Graham, Ian (2008) Requirements Modelling and Specification for Service Oriented Architecture, 1st edition, John Wiley & Sons.

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:

1. Compare/contrast the different ways of camera synthesis.(A2)
2. Explain the rendering process of complicated 3D models using primitives. (A3)
3. Create complicated animation behaviours using 3D transformations. (A4, A7)

Skills

This module will call for the successful student to demonstrate:

4. Critically analyse and manipulate 3D models. (B3, C3, D2)
5. Ability to develop and import 3D object models from different tools to the programme.(B1, D7)
6. Ability to View geometric of 3D models. (B1, B3, C5, D6)
7. Professional treatment of geometric modelling, lighting, and shading. (B5, C1, D7)
8. Professional implementation of collision detection and response techniques. .(B1, B7, C1, D5)
9. Effective verbal presentation of ideas and research skills (C6, D1)

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. (1, 2, 3,4, 8) 7.5%
 - Oral . (1, 2, 3, 4, 6,7, 8) 7.5%
 - Tracing. (1, 2, 3, 4) 7.5%
 - Homework assignments. . (from 1 to 7) 7.5%
 - Lab work and Project. (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.

CS442 Software Construction Quality

Module Code	: CS442
Module Title	: Software Construction Quality
Level	: 4
Credit Points	: 3
Module Leader	: Dr. Ali Fakhry
Pre-requisite	: CS347

Aims

The objective of this module is to acquaint the students with modern principles, techniques, and best practices of software construction. This module focuses on the quality issues pertaining to detailed design and coding, such as reliability, performance, and adaptability. Students will be able to write quality code that is more reliable, reusable, efficient, and/or adaptable to requirements change. This module teaches key software design principles in a studio setting. Students are expected to learn how to build reliable, maintainable, extensible software and how to evaluate other code for those same properties.

Learning Outcomes

Knowledge

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

1. Understand high quality software.(A2)
2. Identify and demonstrate best practices for high quality software such as: data types, high-quality routines characteristics, general issues in using variables and control structures, defensive programming.(A2, A3)
3. Enhance the performance of programme code: code tuning strategies and techniques.(A5)

Skills

This module will call for the successful student to:

4. Demonstrate the ability to construct high quality code and to be able to judge the quality of his colleague's code.(B3, C2, D4)
5. Design new data structures (how to use basic data structures to fit a new problem). (B6, B7, C1, D6)
6. Demonstrate the ability to improve code performance.(B3, C1, C3, D8)

Syllabus

- Product / Process Quality,
 - Software Quality
 - Product Quality
 - Process Quality
 - Process & Process Quality
 - Total Quality Management
 - Software Quality Assurance
- Software Development Methodologies
 - Software Development History
 - Software Development Lifecycle
 - Approaches to Software Development

- Software Development Models
- Software Process Models: ISO9000 / CMM / SPICE
 - Software Quality Assurance
 - Assessment & Improvement
 - Models: ISO9000 / CMM / SPICE
- Software Process Improvement.
 - Software process improvement.
 - Configuration management, Practical realities
 - Project management A structured view

Learning, Teaching and Assessment Strategy

Weekly lectures (3 hours per week): to introduce the basic concepts of the course subjects listed in the syllabus part.

Weekly computer laboratory (1.5 hours per week): to apply the concepts learned to develop workable programming solutions for different types of problems from a variety of fields.

Weekly tutorials (1.5 hours per week): 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.

Teams Project: The students are expected to use a programming language to solve different types of problems from a variety of fields; the lab focuses on assessing the practical skills described earlier. They are expected to do a group project of sizable programming task. All lab work and projects will be assessed on the students, programming ability and efficiency; speed of development, proper use of language constructs proper structure of the programme, clarity and annotation of the programmes and the quality of the overall solution developed; ease of use and speed or execution. All lab assignment and projects should correctly run, be documented and presented.

Assessment

- Unseen Examinations to assess (1 to 5)
- Coursework
 - In Class exams (1,2, 4 & 6)
 - Project defence to assess (1 to 6)

Assessment Weighting

- | | |
|-----------------------|-----|
| • Unseen Examinations | 60% |
| • Coursework | 40% |
| • In Class exams | 20% |
| • Project defence | 20% |

Learning materials

- Continuous Integration: Improving Software Quality and Reducing Risk (Addison-Wesley Signature Series) by Paul Duvall, Steve Matyas, and Andrew Glover (Jul 9, 2007)
- Steve McConnell. Code Complete: A Practical Handbook of Software Construction. 2nd Edition. Microsoft Press, 2007.
- Bertrand Meyer. Object-Oriented Software Construction, 2nd Edition, Prentice-Hall PTR, 2006.

CS444 Embedded Systems

Module Code: CS444
Module Title: Embedded Systems
Level: 4
Credit points: 3
Module Leader: Dr. Ali Fakhry
Pre-requisite: CS102

Aims

This module is concerned with building software for embedded systems using a single microcontroller. The processors examined in detail are from the 8051 family, and programming techniques are going to be using the 'C' language.

Learning outcomes

Knowledge

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

1. A good understanding software for single-processor embedded applications based on small, industry standard, microcontrollers;(A1, A3)
2. Describe the above designs using a modern, high-level programming language ('C'), (A5, A6)
3. Explain embedded "C" / Intel 8051 family microcontrollers' issues.(A1, A5)

Skills

This module will call for the successful student to demonstrate:

4. Develop microcontrollers and how it is functioning with emphases on Intel 8051 family microcontrollers.(B1, B6, B7, C1, C2, D1, D3)
5. Strong programming capabilities using the "C" language with the "Embedded C" ISO extension.(B3,C6, D6)
6. The ability to apply embedded "C" programming to control Intel 8051 family microcontrollers.(B3, C5, D8)

Syllabus

- What are Embedded Systems?
- Introduction to Embedded Systems.
- Embedded Systems Development environment.
- Microprocessors Architecture.
- Programming Embedded Systems.
- Programming Switches Ports & Pins
- Programming Hardware timers
- Creating Embedded Operating System.

- Multi-State Systems
- Programming Serial Interfaces.
- Individually design projects.

Learning, Teaching and Assessment Strategy

Weekly lectures (3 hours per week): to introduce the basic ideas of the course subjects.

Weekly computer laboratory (1.5 hours per week) to use simulation tools to apply the concepts learned in the course.

Team Projects The student will work as a member of project team to apply the concepts learned in the course to analyze, design and programming an application for Intel 8051 microcontroller family.

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 live demonstration of its application.

Assessment

- Unseen examinations (to assess 1,2,3,4,5 & 6)
 - Coursework
 - Two in class exams (to assess 1,3,4 & 5)
 - Two lab work Assignments (to assess 2 & 6)
 - Team Project Defense (to assess 1,2,3,4,5 & 6)

Assessment Weighting

- | | |
|----------------------------|-----|
| • Unseen examinations | %60 |
| • Coursework | %40 |
| • Two in class exams | %10 |
| • Two lab work Assignments | %10 |
| • Team Project Defense | %20 |

Learning materials

Hardware Requirements.

- MicroTRAK Training Kits for 8051

Software Requirements

- Keil C51 Compiler
- Keil 8051 Simulator

Useful Websites

- Berkeley Design technology, Inc.: <http://www.bdti.com>

- EE Times Magazine: <http://www.eet.com/>
- Linux Devices: <http://www.linuxdevices.com>
- Embedded Linux Journal: <http://embedded.linuxjournal.com>
- Embedded.com: <http://www.embedded.com/>
- Embedded Systems Programming magazine
- Circuit Cellar: <http://www.circuitcellar.com/>
- Electronic Design Magazine: <http://www.planetee.com/ed/>
- Electronic Engineering Magazine: <http://www2.computeroemonline.com/magazine.html>
- Integrated System Design Magazine: <http://www.isdmag.com/>
- Sensors Magazine: <http://www.sensorsmag.com>
- Embedded Systems Tutorial: <http://www.learn-c.com/>
- Collections of embedded systems resources
- <http://www.ece.utexas.edu/~bevans/courses/ee382c/resources/>
- <http://www.ece.utexas.edu/~bevans/courses/realtime/resources.html>

Newsgroups

- comp.arch.embedded, comp.cad.cadence, comp.cad.synthesis, comp.dsp, comp.realtime, comp.software-eng, comp.speech, and sci.electronics.cad

Reference Text:

- Embedded C by Michael J Pont", Published by Addison-Wesley Professional (May 4, 2008)
- Programming Embedded Systems in C and C ++ by Michael Barr, Published by O'Reilly, 2007.
- An Embedded Software Primer by David E. Simon, Published by Addison-Wesley, 2008.

CS455 Big Data and Analytics

Module Code: CS455
Module Title: Big data and Analytics
Modul Level: 4
Credit points: 3
Module Leader: Prof. Ali El-Bastawissy
Pre-requisite: CS215

Aims

This module is aimed at getting students up to scratch on Big Data, Hadoop, other NoSQL DBMSs and Multi-Platform Analytics. What is Big Data? How to make use of it? How does it fit within a traditional analytical environment? What skills do students need to develop for Big Data Analytics? This module defines big data and looks at business reasons for wanting to make use of this new area of technology and add the concepts of big data analytics.

Learning outcomes

Knowledge

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

1. Demonstrate What Big Data is. (A2, A3)
2. To clearly understand business use cases for different Big Data technologies. (A4)
3. Classify the technical design issues involved in big data analysis. (A6)
4. Characterize the operating and managerial issues involved in accessing and manipulating Big Data. (A5)
5. Describe Big Data analytical techniques and front-end tools. (A7)

Skills

This module will call for the successful student to:

6. Set up and organise Big Data projects including skills. (B1, B4, B5, C5, C6, D8)
7. Analyse un-modelled, multi-structured data using Hadoop, MapReduce and Sparkmake
8. Use of Big Data to deliver business value (C2, C4, D3, D6, D8)

Syllabus

- Big Data technology platforms beyond the data warehouse
- Big Data analytical techniques and front-end tools
- Big Data platforms and storage options
- Tools and techniques for analysing big data
- Industry use cases - Popular big data analytic applications

Learning, Teaching and Assessment Strategy

- Weekly lectures (3 hours per week) to introduce the basic ideas of the course subjects.
- Weekly computer laboratory (1.5 hours per week) to use Big Data Platforms and Analytical tools and to solve practical case studies. Project: students will work in teams to pursue further studies and hands on Big Data and 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 (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 (6 to 8)

Assessment Weighting

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

Learning materials

Reference Text:

Big Data: Principles and best practices of scalable real-time data systems, by Nathan Marz , James Warren , 1st Edition, ISBN-10: 1617290343

- Data Warehousing in the Age of Big Data, Krish Krishnan, June 2013, The Morgan Kaufmann Series.

CS458 Software Implementation

Module Code:	CS458
Module Title:	Software Implementation
Level:	4
Credit points:	3
Module Leader:	Dr. Ali Fakhry
Pre-requisite:	CS314

Aims

This module discusses and investigates the methods and techniques for testing and maintaining software systems. It builds on the student knowledge of software analysis and design acquired in the prerequisites to present a detailed treatment of software system testing. The experimental part of the module includes software specification and the use of test plan and verification and validation techniques to test the software and record findings.

Learning outcomes

Knowledge

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

1. Illustrate the different testing phases and techniques.(A2, A3)
2. Validate the concepts of consistency and completeness.(A5)
3. Explain the difficulties with Experiments and Validation.(A5)
4. Explain change management and configuration management technologies.(A7)

Skills

This module will call for the successful student to:

5. Design code reading and structured walkthrough for software testing and validation.(B1, C1, D2)
6. Design a test plan: test plan generation, acceptance testing, unit testing, integration testing, and regression testing.(B4, C3, D6)
7. Implement the proof of correctness techniques.(B7, C1, D8)
8. Implement Change management and Configuration management technologies.(B1, B7, C6, D7)

Syllabus

- Manual Software Testing
 - Testing Methods.
 - Testing Development Phases.
 - Testing Operational Activities.
 - Usability Testing.
 - Testing Cycle.
 - Test Cases, Suites, Scripts, and Scenarios
 - Defect Tracking.
 - Test Plan.
 - Test Specification
 - Testing approach
- Automatic Software Testing
 - Automation Testing Basics

- Driving Your Programme
- Results Verification
- Test Automation Planning and Implementation
- Validation and verification.
 - Semantics: Terminology, Taxonomies, and Definitions
 - A Methodology for Accuracy Verification of Codes
 - Error Estimation for Quantification of Uncertainty; Verification of Calculations
 - Systematic Grid Convergence Studies and the Grid Convergence Index (GCI)
 - Applications of Systematic Grid Convergence Studies and the Grid Convergence Index
 - Single Grid Error Estimators
 - Hard Stories
 - Difficulties with Experiments and Validation
 - Methodologies and Examples of Validations, Calibrations, and Certifications
- Code Quality Assurance and Certification
- Change Management technologies.
- Configuration Management Technologies.

Learning, Teaching and Assessment Strategy

Weekly lectures (3 hours per week): to introduce the basic concepts of the course subjects listed in the syllabus part.

Weekly tutorials (1.5 hours per week): The student will be assigned a weekly programming homework to develop on his own. All assignments have to be submitted to the instructor.

Weekly computer laboratory (1.5 hours per week): to apply the concepts learned to develop workable programming solutions for different types of problems from a variety of fields.

Teams Project: The students are expected to use a programming language to solve different types of problems from a variety of fields; the lab focuses on assessing the practical skills described earlier. They are expected to do a group project of sizable programming task.

Assessment

- Unseen Examinations comprehensive exams to assess (1 to 6)
- Coursework
 - In Class exams one or two question exam after the completion of each module section to assess (1,2, 4,6 & 8)
 - Project defence to assess (1 to 8)

Assessment Weighting:

- | | |
|-----------------------------|-----|
| • Unseen Examinations | 60% |
| • Coursework | 40% |
| • In Class exams | 20% |
| • Project defence to assess | 20% |
| • | |

Learning Material

Reference Text

- Pragmatic Software Testing: Becoming an Effective and Efficient Test Professional by Rex Black (Feb 20, 2007) .
- Software Testing: A Craftsman's Approach, Third Edition by Paul C. Jorgensen (Feb 15, 2008)
- Implementing Automated Software Testing: How to Save Time and Lower Costs While Raising Quality by Elfriede Dustin, Thom Garrett, and Bernie Gauf (Mar 14, 2009).
- Practice Standard for Project Configuration Management by Project Management Institute

(Paperback - Jan 31, 2007).

- The Theory and Practice of Change Management: Second Edition by John Hayes (Paperback - Jan 23, 2007).

CS465 Software Project Management

Module Code:	CS465
Module Title:	Software Project Management
Level:	4
Credit points:	3
Module Leader:	Dr. Moustafa Elazhary
Pre-requisite:	CS314

Aims

This module provides a comprehensive coverage of the areas of software project management. Several management aspects are covered such as risk identification and management, and quality management. Testing strategies are also covered. Using automated tools is stressed throughout the course. Upon completion of this course, a thorough understanding of above mentioned areas, and the ability to plan and control a software development project using automated application tools should be gained by the students.

Learning outcomes

Knowledge

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

1. Understand project management techniques such as SDLC and Agile.(A1, A2)
2. Identify the details of building a WBS and user stories for software projects (A5)
3. Identify the required resources, assign them to activities and estimate their cost and cost tradeoffs.(A7)

Skills

This module will call for the successful student to:

4. Assess a given project risks and plan methods for risk management responses.(B3, C1, C3, D2)
5. Professionally manage a software development project.(B2, C6, D1)
6. Develop software project plan.(B7, D6)
7. Implement the software project plan in a real project.(B7, D3)
8. Monitor and control the software development project.(B1, C2, D2)

Syllabus

- The nature of a project: project definition, goals and scope
- Definition of project management: lifecycle, quality and risk
- Project planning: activities; work breakdown structure (WBS), estimating activities' duration, resources and cost
- Resources Estimation.
- Project scheduling: Project network diagram, levelling resources
- Controlling and monitoring projects: progress reporting and evaluation and controlling change
- Managing risk and quality
- Application to information systems

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 use automated project management tools to apply the course concepts to given project-case-studies. Students will also simulate analysis and programming tasks to train on estimating task duration and cost.

Team Projects The student will work as a member of project team to build a moderate size software project. The project will use previously learned programming languages and software development methodologies. It will stress the identification and utilization of the project management techniques for planning and then monitoring the implementation of the project.

Assessment

- The examination questions will be divided equally between testing the student understanding of the concepts introduced (knowledge outcomes) and the applications of this knowledge to practical cases (skills outcomes).
- In class exam (1,2,3 & 4)
- Lab work: The students are expected to do weekly individual computerized assignment applying the project management concepts learned in case studies to assess the skills outcome mentioned above. (5,6,7 & 8)
- Group project: The students are expected to participate in a group project to apply and assess the skills outcomes in sizable assignment. The projects will be judged on the basis of good and accurate planning, project documentation, and the student ability to identify risk and quality issues of their projects. Nevertheless, all projects should be presented. (1,2,3, 4, 5,6, 7 & 8)

Assessment Weighting

- Unseen examinations %60
- Coursework %40
 - Two Lab work assignment %10
 - Two in class exams %10
 - Project defence to assess %20

Learning materials

Software Requirements

- Primavera or MS-Project

Useful Websites

- <http://www.comp.glam.ac.uk/pages/staff/dwfarthi/projman.htm>
- <http://www.systemcorp.com/>

Essential

- Managing A Programming Project: Processes and People, by Philip W. Metzger, John Boddie, Prentice Hall, 2007.

Supplementary Readings

- Software Project Management: A Unified Framework, by Walker Royce, Addison-Wesley, 2008.
- IT Project Management: On Track from Start to Finish, by Joseph Phillips, McGraw-Hill Osborne, 2007.
- Project Management: A Systems Approach to Planning, Scheduling, and Controlling, 8th ed. by Harold Kerzner, John Wiley & Sons, 2007.

CS466 Decision Support and Intelligent Systems

Module Code:	CS466
Module Title:	Decision Support and Intelligent Systems
Level:	4
Credit points:	3
Module Leader:	Prof. Ali El-Bastawissy
Pre-requisite:	CS372

Aims

This module is devoted to introduce decision support systems; show their relationship to other computer-based information systems, demonstrate DSS development approaches, and show students how to utilize DSS capacities to support different types of decisions. Students will analyze, design and implement a managerial decision support system using current development tools.

This module covers the following topics: Introduction to decision support systems; DSS components; Decision making and DSS; DSS software and hardware; developing DSS; DSS models; types of DSS; group DSS; executive information systems; knowledge bases and intelligent systems.

Learning outcomes

Knowledge

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

1. Critically appreciate the basic components, properties and functions of decision support and expert systems. (A1, A3)
2. Understand and explain criteria for the evaluation of decision support and expert systems. (A4, A7)
3. Discuss the different concepts and theories relevant to decision support and expert system. (A5, A7)

Skills

This module will call for the successful student to:

- Analyse an organization's current information and its information needs to identify its DSS/Expert System requirements. (B2, B3, C3, C6)
- Analyse unstructured business problems and select relevant models to aid in their solution. (B7, C1)
- Improve student ability to plan, deep think, create solutions, and brainstorming. (B4, B5, D1, D3)
- Use various support system generators to build elementary decision support systems (B1, B5, C2, D6)
- Work in group in order to participate in a group decision case analysis and solving and reporting (B7, C7, D3)

Syllabus

- Meaning of decision support systems and discussion of various types of DSS
- Data sources, databases, data warehouses, spread sheets.
- DSS components
- Decision making and DSS

- DSS models
- DSS software and hardware
- Developing Models to support decisions making.
- Group decision support systems.
- Executive information systems; artificial intelligence and expert systems.

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 Use various support system generators to build elementary decision support systems.

Class presentations the student will be assigned a specific subject to investigate in depth and make a class presentation.

Assessment

- Exam questions (to assess 1 – 6).
- The class presentation (to assess 4 & 8). It also tests the students' ability for independent learning.
- Lab work: This is designed to assess the practical skills (to assess 4 - 8) of the students. All projects should correctly run, be documented and presented.

Assessment Weighting

- | | |
|---------------------------------------|-----|
| • Unseen Examinations | 60% |
| • In class assessment | 20% |
| • Lab Projects and class presentation | 20% |

Learning materials

Useful Websites

- <http://www.decisionsciences.org/>
- <http://faculty.fuqua.duke.edu/daweb/dafield.htm>

Reference Texts:

- Decision Support Systems and Intelligent Systems/ 7th Ed. Efraim Turban and Jay E. Aronson; Prentice-Hall, 2005.
- Business Intelligence Roadmap: The Complete Project Lifecycle for Decision-Support Applications, by Larissa T. Moss, Shaku Atre, Addison Wesley Professional, 2003.

Supplementary Readings:

Decision Support: An Examination of the DSS Discipline By David Schuff, David Paradice, Frada Burstein, Daniel J. Power, Ramesh Sharda, Springer 2011.

CS472 Advanced Database Systems

Module Code: CS472
Module Title: Advanced Database Systems
Level: 4
Credit Points: 3
Module Leader: Prof. Ali El-Bastawissy
Prerequisites: CS215

Aims

This module expands the principals of database management systems introduced in the prerequisite module to provide more advanced topics that cover a broad range of concepts, modelling, and system implementation techniques. The focus will be: on illustrating how issues such as database catalogue, query processing, transaction processing, security, concurrency controls are implemented in real database management systems, and on emerging technologies such as object-oriented database models and system implementation techniques.

Learning Outcomes

Knowledge

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

1. Critically appreciate the advanced concepts of database management system. (A1, A2)
2. Understand the object orient database models. (A4)
3. Demonstrate the techniques of implementing transaction processing in a database environment. (A5)

Skills

This module will call for the successful student to:

4. Critically analyse the important issues and techniques of database security. (B3, B7, C3, D2)
5. Evaluate the suitability of object oriented data models for a given application. (B1, B3, D6)
6. Compare, evaluate and setting up the real RDBMSs (B5, C2, D6)
7. Apply advanced techniques such as query optimization and transaction processing in database applications. (B1, C6, C2, D8)
8. Manage database security, authentication and authorization. (B7, C2, C7, D4)

Syllabus

- Advanced topics in SQL
- Examples of Relational Database Management Systems: Oracle
- Database System Architecture and the System Catalogue
- Query Processing and Optimization
- Transaction Processing Concepts
- Concurrency control
- Recovery
- Database Security and Authorization
- Concepts for Object-Oriented Database Systems

- Overview of the Object Model of ODMG and the Object Definition Language (ODL)
- The Object Query Language (OQL)
- Object Relational and Extended Relational Database Systems
- Emerging Database Technologies and Applications

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

Term Paper: The student will work individually to prepare a short paper on one of the advanced database subjects discussed in module. Some of the better paper will be presented in class.

Assessment

Assessment will be based on:

- two unseen exams composed of several questions to assess the student knowledge and understanding (1 to 8)
- Case studies and assignments are used to assess (3,5,6,7,8) 20%
- Presentation and term paper to assess (1,2,4) 20%

Assessment Weighting

- Unseen examination 60 %
- Coursework (no examination) 40 %
 - Stored Procedures and Triggers 20%
 - Presentation and term paper 20%

Learning materials

Essential

- Fundamentals Of Database Systems, 6th ed. by Ramez Elmasri and Shamkant B. Navathe, Addison-Wesley Copyright: 2011.

Recommended Readings

- An Introduction to Database Systems, 8th ed. by C. J. Date, Addison-Wesley, 2009.
- Database System Concepts by Abraham Silberschatz, Henry F. Korth, and S. Sudarshan McGraw-Hill- Aug 1, 2007.
- Database Systems: Design, Implementation, and Management by Peter Rob and Carlos Coronel, Course Technology, 2008.

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 analysing large databases using the appropriate software.

Learning outcomes

Knowledge

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

1. Demonstrate the common data mining techniques.(A1)
2. Illustrate the use and expected outcomes of applying data mining to different data sets. (A3, A4)
3. Demonstrate the theory and algorithms used in data mining models (A5)
4. 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 (A7)

Skills

This module will call for the successful student to:

5. Analyse large volumes of data using such technologies as: Machine learning, Statistics, Pattern Recognition, Artificial Intelligence, and Database Systems. (B3, B7, C1, D5, D8)
6. Develop appropriate models for data mining (B3, B5, C2, D4, D7)

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 (2, 3, 5) 60%
- In Class Assessment (2, 3, 5) 20%
- Lab Project Assessment (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.

CS476 Web Database Application

Module Code: CS476
Module Title: Web Database Application
Level: 4
Credit Points: 3
Module Leader: Dr. Moustafa M. Elazhary
Prerequisites: CS385

Aims

Students will learn how to exploit the conceptual modelling approach of software engineering, from idea to application. They will learn to harness the design technologies of relational databases for use on the web, and to transform their conceptual designs of data-intensive web applications into effective software components. The applicability of these techniques to structured and unstructured data intensive sites is important, as long as there is structured metadata associated with the unstructured data.

Learning Outcomes

Knowledge

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

1. Evaluate database technologies and techniques for web applications (A1, A2)
2. Implement the querying and reporting methodologies over the web (A4)

Skills

This module will call for the successful student to:

3. Design the data-intensive web application (B1, C6, D2)
4. Develop data-intensive web application (B3, C5, D6)
5. Analyse web database solutions (B5)
6. Formulate and contrast advanced SQL (B3, D7)
7. Assess your database function (B3, D6, D8)
8. Compare a three-tier with single and two-tier web applications (B5, C6)

Syllabus

- Web database programming environment
- Web database programming technologies and techniques
- Writing stored procedures for the web application
- Web queries and reports
- Refactoring SQL Applications
- Database and XML
- Web database transaction processing
- Creating a three-tier web Application

Learning Teaching and Assessment Strategies

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

Weekly computer laboratory to develop data-based Web application.

Team Projects 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 data-based web programming.

Assessment

Unseen exams: Two unseen exams several questions to assess the student knowledge and understanding (1, 2, 3, 5, 6)

Course Work: composed of Assignments, Lab work and team project:

Lab weekly progress to assess (2, 3, 5, 6, 7)

Project defence to assess (3, 4, 5, 8)

Assessment Weighting

- Unseen Examinations 60%
- Coursework 40%

Learning Material

Reference Text

- Welling, Luke (2008) PHP and MySQL Web Development, 4th edition, Addison Wesley

Supplementary Readings

- Ceri, Stefano (2003) Designing Data-Intensive Web Applications, 1st edition, Morgan Kaufmann
- Faroult, Stephane (2008) Refactoring SQL Applications, 1st edition, O'Reilly
- Feuerstein, Steven et al (2006) MySQL Stored Procedure Programming, 1st edition, O'Reilly
- Henderson, Ken (2002) The Guru's Guide to SQL Server Stored Procedures, XML and HTML, 1st edition, Addison Wesley

CS477 Machine Learning

Module Code:	CS477
Module Title:	Machine Learning
Level:	4
Credit points:	3
Module Leader:	Dr. Soha Safwat
Pre-requisite:	CS361

Aims

The main objective of this module is to design and implementation of learning programmes that improve their performance on some set of tasks with experience. Upon successful completion of the course, students will have a broad understanding of machine learning algorithms and their use in data-driven knowledge discovery and programme synthesis. Students will also be able to identify, formulate and solve machine learning problems that arise in practical applications. Students will have a knowledge of the strengths and weaknesses of different machine learning algorithms (relative to the characteristics of the application domain) and be able to adapt or combine some of the key elements of existing machine learning algorithms to design new algorithms as needed.

Learning outcomes

Knowledge

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

1. Introduce the fundamental problems of machine learning(A1)
2. Provide understanding of techniques, mathematical concepts, and algorithms used in machine learning to facilitate further study in this area.(A3)
3. Provide understanding of the limitations of various machine learning algorithms and the way to evaluate performance of machine learning algorithms. (A5)
4. Practice software implementation of different concepts and algorithms covered in the course(A2)

Skills

This module will call for the successful student to:

5. To be able to assess the strengths and weaknesses of several machine learning algorithms. (B3, C2, D2)
6. Use machine learning concepts to solve problems. (B3, C5, D4)
7. Implement different machine learning to perform various tasks in different application domains(B7, C6, D6)

Syllabus

- Overview of machine learning
- Non-parametric learning
- Bayesian learning
- Neural Networks
- Data and dimensionality reduction
- Ensemble learning

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

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

Learning materials

Essential

Machine Learning, E. Alpaydin, MIT Press, 2010.

Recommended

- Pattern Recognition and Machine Learning, C. Bishop, Springer, 2006.
- Pattern Classification, R. Duda, E. Hart, and D. Stork, Wiley-Interscience, 2000.
- Machine Learning, T. Mitchell, McGraw-Hill, 1997.

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:

1. Differentiate between the different scientific fields involved in interaction design. (A2, A6)
2. Illustrate the principles and the applications of ID design goals, usability goals, user experience etc. (A2)
3. Analyse how much the theories of how people communicate and work can influence the design of interactive systems. (A4)
4. Illustrate the different methodologies used in interface design and users involvement. (A6, A7)

Skills

This module will call for the successful student to:

5. Select models that are appropriate to particular design problems and contexts and justify those choices. (B1, B2, C6, C7, D5)
6. Apply standard usability evaluation techniques to evaluate and critique designs from a usability perspective, and to propose improvements. (B5, B6, D4, D6)
7. Design interactive systems that are usable and meet the users' needs. (B7, C6)

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 (1 to 7)
- Lab case studies: to assess (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.

CS485 Pattern Recognition

Module Code: CS485
Module Title: Pattern Recognition
Level: 4
Credit points: 3
Module Leader: Dr. Soha Safwat
Pre-requisite: CS361

Aims

This course focuses on the underlying principles of pattern recognition and on the methods of machine intelligence used to develop and deploy pattern recognition applications in the real world. Emphasis is placed on the pattern recognition application development process, which includes problem identification, concept development, algorithm selection, system integration, and test and validation.

Learning outcomes

Knowledge

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

1. Introduce the student to the basic concepts and methods for the recognition of patterns in data. (A1)
2. Provide the student with a working knowledge of the pattern recognition application development process. (A2)
3. Understand the concept of a pattern and the basic approach to the development of pattern recognition algorithms (A5)
4. Understand and apply methods for pre-processing, feature extraction, and feature selection to multivariate data (A7)

Skills

This module will call for the successful student to:

5. Apply both supervised and unsupervised classification methods to detect and characterize patterns in real-world data. (B3, C2, D2, D4,)
6. Develop prototype pattern recognition algorithms that can be used to study algorithm behaviour and performance against real-world multivariate data. (B6, B7, C6, D7, D8)

Syllabus

- Introduction to patterns and pattern recognition application development
- Supervised pattern detection I
- Feature extraction - multivariate data
- Feature extraction - image data
- Supervised pattern detection II (linear classifiers)
- Unsupervised pattern detection I (clustering)
- Supervised pattern detection IV (rule-based classifiers)
- Sensor and data fusion

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% (1,2) |
| • Lab Projects | 20% (2,3) |
| • Unseen Examinations | 60% (3,4,5,6) |

Learning materials

Essential

Pattern Recognition 4/e, Theodoridis / Koutroumbas, Academic Press, 2009.

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:

1. Conceptualize image processing problems. (A1, A2)
2. Demonstrate the image processing concepts. (A4)
3. Illustrate the standard image processing algorithms.(A5)
4. Compare/contrast the different applications of computer vision. (A5)

Skills

This module will call for the successful student to:

5. Apply the concepts and methodologies for the formation, representation, enhancement and analysis of digital images. (B3, C5, D4)
6. Use available tools to develop applications of image processing. (B1, C2, D2)
7. Improvise the design and implementation of image processing algorithms to suite specific applications. (B2, C1, D7)
8. Using benchmark images for algorithm testing.(B7, C6, D8)

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 (1 to 6)
- In class assessment formative discussion to assess (1 to 6)
- Project is used to assess all outcomes (1 to 8) the project will be evaluated on 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.

CS488 Robotic Interfacing

Module Code: CS488
Module Title: Robotic Interfacing
Level: 4
Credit Hours: 3
Module Leader: Dr. Magda El-Daghestany
Prerequisites: CS203

Aims

This module introduces the student to the multidisciplinary growing field of robotics. Different robot architectures and computer interfacing techniques are discussed, to apply computer controlled robotic applications. Other topics, including path planning and robot sensing, open a very wide range of applications in other modules (graduation project).

Learning Outcomes

Knowledge

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

1. Differentiate between different robotic architectures and interfacing techniques. (A1, A2)
2. Illustrate theories related to robotic motion and path planning. (A1, A2, A3)
3. Analyse different problems related to robotics. (A7)

Skills

This module will call for the successful student to:

1. Choose theories and models that are appropriate to a particular design problem. (B3, D2, D7)
2. Apply computer controlled robotic applications using the appropriate software. (B7, C1, C5)
3. Conduct independent literature review and research in a specific topic. (D1, D3, D5)

Syllabus

- Different robotic architectures and basic parts.
- Computer interfacing and robots.
- Computational problems in robotics.
- Robot sensing.

Learning Teaching and Assessment Strategies

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 apply requirement gathering and specification methods, interface design models and evaluation methods

Research Paper and Class Presentation: The student will choose a subject from the module material, conduct a thorough literature search on the subject and prepare a research paper.

Then his work is introduced in a presentation to critically discuss his research topic.

Assessment

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

The exam questions will be divided equally between assessing the knowledge and skill outcomes mentioned above.

Lab Project and assignments: Lab assignments are designed to train students individually in practical applications. A lab project assesses the students' practical skills mentioned above

Research Paper: the students should work in groups to select any subject related to the module and prepare a paper about it. A class presentation is conducted to introduce their work. This is designed to assess the students' knowledge outcomes and their ability to learn independently.

Assessment Weighting

- | | |
|---|-----|
| • Unseen Examinations (1 – 4) | 60% |
| • In class test (1 – 4) | 10% |
| • Research Paper & Presentation (3,4,6) | 15% |
| • Lab Project (3,4,5) | 15% |

Learning Material

Software Requirements

- Relevant software.

Reference Text

- Embedded Robotics: Mobile Robot Design and Applications with Embedded Systems, 3rd ed. Thomas Braunl, Springer, 2008

Additional References

- Robotics, Appin Knowledge Solutions, Infinity Science Press, 2007.

CS489 Semantic Web Programming

Module Code: CS489
Module Title: Semantic Web Programming
Level: 4
Credit Points: 3
Module Leader: Dr. Moustafa M. Elazhary
Prerequisites: CS385

Aims

This module explains to the students how they can make use of semantic programming techniques to enrich and simplify web applications. It covers the common formats for integration and combination of data drawn from diverse sources, as well as the language for recording how the data relates to real world objects.

Learning Outcomes

Knowledge

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

1. Understand the semantic Web technologies (A1)
2. Discuss the languages used in semantic Web (A2)
3. Create semantic Web services (A4)
4. Create semantic Web Applications (A5, A7)

Skills

5. Analyse the component technologies of the Semantic Web and explain their roles (B4, D2)
6. Implement XML and RDF (Resource Description Framework) (B3, C6, D6)
7. Develop a web services application that “discovers” the data and/or other web services via the semantic web (B6, C3, C6, D8)
8. Develop semantic Web applications to solve real-world problems (B7, C5, D8)

Syllabus

- Semantic Web vision
- Semantic Web technologies
- RDF (Resource Description Framework)
- Web Ontology Language: OWL
- Logic and Inference: Rules
- Semantic Web service
- Ontology Engineering
- Creating a semantic Web application

Learning Teaching and Assessment Strategies

Weekly lectures to introduce the basic concepts of the course subjects

Weekly computer laboratory to develop a semantic web application

Team Projects 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 semantic web programming.

Assessment

Two unseen exams several questions to assess the student knowledge and understanding (1, 3, 5, 7)

Assignments, Lab work and team project:

Lab weekly progress to assess (2, 4, 5, 6, 8)

Project defence to assess (2, 4, 6, 7, 8)

Assessment Weighting

- Unseen Examinations 60%
- Coursework 40%

Learning Material

Reference Text

- Antoniou, Grigoris et al (2008) Semantic Web Primer, 2nd edition, The MIT Press.

Supplementary Readings

- Segaran, Toby et al (2009) Programming the Semantic Web, 1st edition, O'Reilly Media, Inc.
- Hebel, John et al (2009) Semantic Web Programming, 1st edition, Wiley.

ENG101x English for Academic Writing Purposes

Module Code: ENG101x

Module Title: English for Academic Writing Purposes

Level: 1

Credit Points: 3

Module Leader: Dr. Aziza Hafez

Prerequisite: ENG 90 or passing the ELAT exam

Aims:

This course is geared towards helping students in effectively writing academic essays and avoiding common errors in writing. In addition, reading passages are used as a means of teaching students reading comprehension, style and organization of writing, summary writing and understanding vocabulary from context.

Syllabus

Introduce the course objectives and the grading system.

- Unit 1: Reading for Writing: "Narrative of the Life of Frederick Douglass" , Prewriting Activities, Structured Writing.
- Unit 2:"What's Your Verdict?" "Fluency Practice: Freewriting, Reading for Writing: "The Case of Leroy Strachan", Prewriting Activities, Structured Writing.
- Unit 3:"Men and Women: Nothing but the Facts"Fluency Practice: Freewriting, Reading for Writing: "The Case of Leroy Strachan" , Prewriting Activities, Structured Writing Focus.
- Unit 4:"The Best Time To Be Alive"Fluency Practice: Freewriting, Reading for Writing: "There's No Time Like the Past", Prewriting Activities, Structured Writing Focus, Structured Writing.
- Unit 5:"The Happiest School in the World"Fluency Practice: Freewriting, Reading for Writing: "Summerhell: A Radical Approach to Child Rearing", Prewriting Activities , Structured Writing Focus, Structured Writing.
- Unit 6:"Are you Getting Enough Sleep"Fluency Practice: Freewriting, Reading for Writing: "The Causes of Sleep Deprivation in America: a Nation of Walking Zombies" , Prewriting Activities, Structured Writing Focus, Structured Writing.
- Unit 8:"For and Against Bilingual Education"Fluency Practice: Freewriting, Reading for Writing: "Bilingual Education: Parents' Views", Prewriting Activities, Structured Writing Focus.
- Unit 9:"Case Studies in Business Ethics: Malden Mills and Ben & Jerry's Ice Cream"Fluency Practice: Freewriting, Reading for Writing: "Malden Mills" , Prewriting Activities, Structured Writing Focus.
- Unit 10:"The Tell-Tale Heart"Fluency Practice: Freewriting, Reading for Writing(Cont.): "The Tell-Tale Heart", Prewriting Activities, Structured Writing Focus,

Assessment Scheme:

Students sit a midterm unseen exam of 90 minutes – with an essay, a summary, a reading comprehension and editing exercises. Students sit a final unseen exam of 3 hours - with an essay, a summary, a reading comprehension and editing exercises. Students present a portfolio compiling all the essays they had written throughout the semester with their different drafts.

Assessment Pattern:

Coursework	40%
Unseen Examinations	60%

Learning Materials:**Textbooks:**

Mary R. Colonna & Judith E. Gilbert. *Reason to Write*. (Intermediate)UK.:Oxford Univ.Press, 2006

Useful Tools :

www.eslcafe.com

www.geocities.com/SoHo/Atrium

www.204.pair.com/ebaack

www.io.com

<http://owl.english.purdue.edu/>

www.better.english.com

www.eviews.net/references.html

www.ohiou.edu/esl/english/index.html

ENG102x English for Study Skills

Module Code: ENG102x

Module Title: English for Study Skills

Level: 1

Credit Points: 3

Module Leader: Dr. Aziza Hafez

Prerequisite: ENG101x

Aims:

This course helps students acquire study skills that would facilitate any research process. Students are also taught types of business writing, such as reports, business letters, memos, and curriculum vitae. Special focus also goes to having students acquire presentation skills through presenting their own reports.

Syllabus:

Unit 1: Business Communication Basics

General introduction to the course 102/126

Unit 2: Speaking, Listening and Non-verbal Communication

Unit 3: Use of English in communication

Introducing the report as a type of business writing

- Brainstorming

- Skimming and scanning

- Bibliography cards (APA methodology)

- Introduction to paraphrasing- Paraphrasing Exercises

- Library skills-Plagiarism

- Report outline

- More applications on paraphrasing

- Quiz on paraphrasing

- In-text citation

- Organizational analysis

- Unit 5: Rules of good writing

Introducing the business letter

- Reply to enquiry- Letter of confirmation

- Acknowledgement Letter

- Enquiry letter

- Collection letters

- Complaint letter

- Reply to a complaint

- Designing a questionnaire

- Application letter & CV as examples of business writing

- Offer of employment letter
- Letter of acceptance
- Letter of resignation

- Fax messages
- E-mail

- Data representation: Tables
- Data representation: Graphs
- Findings, conclusion, and rest of report pages

- Memos
- Oral presentation skills

Learning Outcomes:

Knowledge

- 1- At the end of this module students will recognize the importance of business communication skills and will be able to distinguish between different communication means in business, analyze the texts they need to incorporate in their reports, as well as identify graphical, visual and statistical information.(D2, D4, D5)

Skills

At the end of this module students will be able to:

- 2- paraphrase, summarize and analyze the texts they need to incorporate in their reports.(D5, D8)
- 3- write clear and effective curriculum vitae, business letters, faxes, e-mails and memos.(C4)
- 4- design and administer questionnaires.(D2, D8)
- 5- analyze the quantitative and qualitative data obtained from the questionnaires.(D2)
- 6- integrate graphical, visual and statistical information into their reports.(D5)
- 7- produce a report with an outline and a simplified "Works Cited" page.(C4, D1)
- 8- present their reports using slides or computer software.(D1)

Assessment Scheme:

Students sit a midterm unseen exam of 90 minutes – with short answer questions and several technical writing tasks. Students sit a final unseen exam of 3 hours – with short answer questions and several technical writing tasks. Students produce a report about a current problem or issue. Their report should incorporate background information about the problem and the results of a questionnaire they had designed and administered. They submit a 2000 word report. A presentation of their reports is also required.

Assessment Pattern:

Coursework	40%
Unseen Examinations	60%

Learning Materials:

Text Book:

Shirley Taylor. Communication for Business: A Practical Approach. 4th ed. Longman, 2005.

Useful Tools:

Search engines

www.yahoo.com

www.google.com

www.altavista.com

www.ipl.org

Useful links for business and report writing:

www.devry-phx.edu/lrnresrc/dowsc/

owl.english.purdue.edu/

www.io.com

www.better.english.com

ENG201x English for Research Purposes

Module Code: ENG201x

Module Title: English for Research Purposes

Level: 2

Credit Points: 3

Module Leader: Dr. Aziza Hafez

Prerequisite: ENG101x & ENG102x

Aims:

This course emphasizes research skills necessary for writing research papers. It also provides a survey of different articles on specialized topics. The course trains the students on rhetorical awareness beyond traditional composition, intensive writing practice with a thorough guidance on using references and citing sources.

Syllabus

Introduction to the course (Objectives – assignments – grading)

Introduction of selected research topics

Chapter One: **Writing from Research**

Chapter two: **Finding a Topic** (All except pp. 24 – 31)

Library Skills: Chapter Three: Finding and Filtering Electronic Sources:

Library Skills: Chapter Four: **Gathering Data in the Library**

Chapter Six: **Understanding and Avoiding Plagiarism**

Chapter Seven: **Finding and Evaluating Sources**

Definition + Process Writing Approaches

Practice (Analysis and Production Tasks)

Practice (Analysis and Production Tasks) cont.

Comparison and Contrast + Division and Classification

Practice (Analysis and Production Tasks)

“Expressing a Thesis Sentence, Enthymeme, and Hypothesis” (Chapter Two p. 24) + Exercises

Chapter Eight: “Writing a Rough Outline” + “Writing a Formal Outline”

For MLA Classes

Chapter 9: **Writing Effective Notes**

For APA Classes

Chapter 5: **Conducted Research Outside the Library**

For MLA Classes

Chapter 9 (cont.)

For APA Classes

Chapter 9: **Writing Effective Notes**

Argumentation and Persuasion + Cause and Effect.

Practice (Analysis and Production Tasks).

Drafting the Paper in an Academic Style.

Writing the Introduction, Body and Conclusion.

Revising, Proofreading, and Formatting the Rough Draft.

Fallacies.

Learning Outcomes

Knowledge:

- 1- At the end of this module students will be able to analyze different texts to identify thesis statements and developmental functions of those texts, identify fallacies in the texts they analyze, identify different library classification systems and card catalogue.(D4, D8)

Skills:

At the end of this module students will be able to:

- 2- write outlines and summaries.(D1)
- 3- develop logical arguments.(D2)
- 4- write a research paper using correct in-text citations according to the MLA style.(C4, D5)
- 5- prepare in their research paper a complete 'Works Cited' or 'References' page prepared according to the MLA or APA style.(C4, D4, D5)
- 6- present their papers using slides or computer software.(D1)

Assessment Scheme:

Students sit a midterm unseen exam of 90 minutes – with short answer questions and several writing tasks. Students sit a final unseen exam of 3 hours – with short answer questions and several writing tasks. Students produce a research paper that incorporates information from different sources. They must demonstrate an understanding of the topic they select, and develop a logical argument. Their paper should follow correct research skills such as in-text citation and works cited. They submit a 6000 word paper. A presentation of their papers is required as well.

Assessment Pattern:

Coursework:	40%
Unseen Examinations	60%

Learning Materials:

Textbook:

James Lester. Writing Research Papers. 12th ed. Pearson, Longman, 2007

Useful Tools:

Internet search engines:

www.yahoo.com

www.google.com

www.altavista.com

www.ipl.org

Useful links for research writing:

<http://www.devry-phx.edu/lnresrc/dowsc/>

MTH100 Calculus

Module Code:	MTH100
Module Title:	Calculus
Level:	1
Credit points:	3
Module Leader:	Dr. Magda El-Daghestany
Pre-requisite:	None

Aims

This module aims to introduce basics of Calculus. The concepts and rules of differentiation of functions, including algebraic, trigonometric and exponential functions, discussed. Basics of integration are also introduced.

Learning Outcomes

Knowledge

After completing this module students will be able to:

- Identify real numbers and real line coordinates.(A1)
- Understand functions, continuity and derivation of the function.(A1)
- Understand chain rule and implicit differentiation.(A1)
- Understand integration basic rules.(A1)

Skills

This module will call for the successful student to:

- Test continuity and discontinuity.(B3, D2, D4, D8)
- Apply implicit differentiation and the chain rule.(B3, D2, D4, D8)
- Differentiate and integrate different functions.(B3, D2, D4, D8)

Syllabus

- Real Numbers and the Real Line, Coordinates, Lines, and Increments.
- Functions, Shifting Graphs, Trigonometric Functions, Exponential and Logarithmic functions.
- Rates of Change and Limit Rules.
- Continuity, Tangent Lines, The Derivative of a Function and Differentiation Rules.
- Applications on Derivatives.
- Graphing with Limits as Asymptotes, and Dominant Terms.
- Indefinite Integrals, Integration by Substitution and Definite Integrals.

Learning, Teaching and Assessment Strategy

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

Weekly tutorials to apply the concepts introduced in the lectures and discuss the solution of the homework assignments.

Assessment

Homework Assignments for feedback. (1 - 4)

In class tests. (5 - 7)

Unseen examinations: to test the student knowledge and understanding of the material delivered. (1 - 7)

Assessment weighting

- Unseen Examinations 60 %.
- Coursework 40%.

Learning materials

Essential

- Thomas' Calculus Early Transcendentals, Twelfth Edition, by George Thomas, Maurice Weir and Joel Hass, Pearson, 2010.

Recommended

- Calculus and Analytical Geometry, by George B. Thomas & Ross L. Finney, Addison-Welsey, 1996.
- Calculus Labs using Mathematical, by Arthur G. Sparks, John Davenport and James Braselton, Harper Collins College Publisher.

MTH103 Discrete Mathematics

Module Code: MTH103
Module Title: Discrete Mathematics
Level: 1
Credit points: 3
Module Leader: Dr. Magda El-Daghestany
Pre-requisite: None

Aims

This module is the basis of Mathematics for Computer Science. It teaches students how to think logically and mathematically. Formal Logic, Set Theory, Inductive Proofs and Counting are introduced, with applications.

Learning Outcomes

Knowledge

After completing this module students will be able to:

- 1- Know the basics of formal logic. (A1, A4)
- 2- Understand induction, recursion and their applications. (A1, A4)
- 3- Know the basics of Set Theory. (A1, A4)
- 4- Understand counting arguments. (A1, A4)

Skills

This module will call for the successful student to:

- 5- Solve formal logic problems. (B3, D2, D4, D8)
- 6- Differentiate between true and false statements as well as prove or disprove them. (B3, D2, D5)
- 7- Use sets and set operations in formal arguments. (B3, D2, D8)
- 8- Build counting arguments. (B3, D2, D4, D5)

Syllabus

- Introduction to Logic.
- Introduction to Proof Techniques.
- Recursive Definitions and Recurrence Relations.
- Sets: Set Operations, and Set Identities.
- Counting sets. Addition and Multiplication Principle.
- Permutations and Combinations.
- Functions.
- Relations.

Learning Teaching and Assessment Strategy

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

Weekly tutorials to apply the concepts introduced in the lectures and discuss the solution of the homework assignments.

Assessment

Unseen examinations: to test the student knowledge and understanding of the material delivered. (1 – 8).

Homework Assignments for feedback. (1 – 4).

In class tests (5 – 8).

Assessment Weighting

- Unseen Examinations 60%
- Coursework 40%

Learning materials

Essential

- Discrete Mathematics and its Applications, Kenneth Rosen, McGraw-Hill, 2013 .

Recommended

- Discrete Mathematical Structures, Kolman, 4th ed., 2000.

MTH106 Linear Algebra

Module Code: MTH106
Module Title: Linear Algebra
Level: 1
Credit points: 3
Module Leader: Dr. Magda El-Daghestany
Pre-requisite: MTH100

Aims

This module is an introduction to the basics of linear algebra. Matrix and vector operations are studied to solve a system of linear equations. The concepts of vector spaces, Eigen values and Eigen vectors are understood and used to solve linear problems.

Learning Outcomes

Knowledge

After completing this module students will be able to:

- 1- Know the basics of matrix theory including matrix properties and functions. (A1)
- 2- Understand the abstract concepts of vector spaces. (A1)
- 3- Know how to apply linear algebra in solving linear systems of equations. (A1)

Skills

This module will call for the successful student to:

- 4- Solve matrix problems. (B3, D2)
- 5- Have the ability to differentiate between true and false statements about matrices and vector spaces as well as prove or disprove them. (B3, D4)
- 6- Use linear algebraic concepts to tackle linear problems. (B3, D2, D8)

Syllabus

- Introduction to Matrices: Matrix Operations and their Properties.
- Solutions of Linear Systems of equations in matrix form.
- Matrix Inverse, Determinants and Cramer's Rule.
- Vector Spaces and Subspaces.
- The Span and Linear Independence.
- Basis and Dimension of Vector Spaces.
- The Rank of a matrix.
- Introduction to Eigenvalues, Eigenvectors and Diagonalization.

Learning Teaching and Assessment Strategy

Weekly lectures: to introduce the basic concepts of the module subjects.

Weekly tutorials: to apply the concepts introduced in the lectures and discuss the solution of the weekly homework assignments

Assessment

Unseen examinations: exam questions assess the students ability to solve linear problems in matrix form and understand the concept of vector space. (1 - 6)

Homework assignments. (1 - 3)

In class tests (4 – 6)

Assessment Weighting

- Coursework 40%
- Unseen Examination 60%

Learning Materials

Useful Websites

- http://www.psc.edu/~burkardt/papers/linear_glossary.html

Reference Text

- Introductory Linear Algebra, by Bernard Kolman and David Hill, Pearson Prentice Hall, 2013.

Supplementary Readings

- Matrix Analysis and Applied Linear Algebra, by C.D. Meyer.

MTH204 Probability and Statistics

Module Code: MTH204
Module Title: Probability and Statistics
Level: 2
Credit points: 3
Module Leader: Dr. Magda El-Daghestany
Pre-requisite: MTH100

Aims

The module aims to introduce analytical methods to solve problems, using concepts from probability and statistics. This includes random variables and their distributions, mathematical expectation, point and confidence interval estimation, correlation and applications.

Learning outcomes

Knowledge

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

- 1- Distinguish between different probability distributions. (A1, A4)
- 2- Analyze different statistical problems. (A1, A4)
- 3- Find problem solution and find confidence intervals. (A1, A4, B3)

Skills

This module will call for the successful student to demonstrate:

- 4- Select the appropriate probability distribution in a specific application. (B3, C3, D2, D8)
- 5- Testing statistical hypothesis. (D2, D8)
- 6- Making statistical decisions. (C3, D2, D4, D5)

Syllabus

- Describing Data.
- Probability and Random variables.
- Mathematical expressions.
- Discrete distribution.
- Continuous distribution.
- Statistical inference.

Learning, Teaching and Assessment Strategy

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

Weekly tutorials to apply the concepts introduced in the lectures and discuss the solution of the homework assignments.

Assessment

Unseen examinations (1 - 6)

Homework assignments for feedback (1 - 3)

In class tests (4 – 6)

Assessment Scheme

- Unseen Examinations 60 %
- Coursework 40%

Learning materials

Reference Text

- Applied Statistics and Probability for Engineers, Fifth edition, Douglas Montgomery and George Runger, John Wiley, 2011.

Supplementary Readings

- Statistical Techniques in Business and Economics, Fourteenth edition, M. Lind, Mc-Graw Hill International edition, 2010.

PHY103x Fundamentals of Electronics

Module Code: PHY103x
Module Title: Fundamentals of Electronics
Level: 1
Credit points: 3
Module Leader: Dr. Nabila Nowaira
Prerequisite: None

Aims:

The module gives the students an idea about semiconductors with their two main devices, namely, pn-junction diodes and transistors. The students learn the main characteristics of these devices. Thereafter, diodes and transistors are used to build electronic circuits which perform different specific functions.

Learning outcomes:

Knowledge:

After completing this course students will be able to:

- 1- Using the laws of electricity to analyse electric circuits.(A1)
- 2- Differentiate between different types of semiconductors.(A1, A3)
- 3- Appreciate suitable conditions to operate diodes and transistors.(A1, A3)
- 4- Construct and analyse electronic circuits of specified functions.(A1, B6)
- 5- Performing measurements to check the faults in electronic circuits.(A1, B6)

Skills:

- 6- Check the electronic components to choose the suitable ones.(D5)
- 7- Use the chosen components to construct electronic circuits of specific functions.(D3, D4)
- 8- Reading the data sheets of the chosen components to define the operating conditions of the components used.(D2, D8)

Syllabus

- Introduction to electric circuits
- Electric current and Ohm's Law
- Elements of Electric circuits
- Kirchhoff's Laws
- Semiconductor diodes:
- Introduce the band theory of solids to differentiate among insulators, conductors and semiconductors.
- Semiconductor diodes; Zener diodes; Light-Emitting diodes (LEDs).
- Diode applications:
- Load-line analysis.
- AND / OR gates.
- Half-wave and Full-wave rectification.
- Clippers, Clampers and Voltage multiplier circuits.
- Bipolar junction transistor: (BJT)
- Transistor construction.
- Transistor operation.

- Common-base configuration.
- Transistor amplifying action.
- Common-Emitter configuration.
- Common-Collector configuration.
- Limits of operation.
- DC Biasing-(BJT):
- Operating point.
- Fixed –bias circuit.
- Emitter-Stabilized bias circuit.
- Voltage-Divider bias.
- DC-Bias with voltage feedback

Lab Experiments

- Determination of unknown resistances using Ohm's Law.
- Verifying Kirchhoff's Laws.
- Determination of the band gap in a semiconductor.
- Diodes and Rectifier circuits.
- BJT characteristics
- BJT as a switch

Learning Teaching and Assessment Strategy

Weekly lectures to introduce the basic items of the course subjects

Bi-Weekly tutorials: to discuss and solve homework assignments.

Bi-Weekly laboratory: to apply practically some of the theoretically taught ideas.

Assessment

- Five class tests (1 – 4).
- Assignments in the form of drill problems concerning the material covered each week. (1-4)
- Lab experiments and projects. (4 - 8)

Assessment weighting

- | | |
|----------------------|-----|
| • Coursework | 40% |
| • Unseen examination | 60% |

Learning Materials

Reference Texts

- “Electronic devices”, Thomas L. Floyd.
- “Electronic devices and circuit theory”, Robert L. Boylestad & Louis Nashelsky

MGT200 Introductory Management

Module Code	MGT200
Module Title	Introductory Management
Level	1
Credit	3
Module Leader	Dr. Emad Elwy
Pre-requisite	ENG101

Aims

This module aims to provide students with a solid grounding in the core concepts and functions of management. It also enables students to develop their practical skills in the study of real world management practice. It also gives students an appreciation of the field of management studies.

Learning Outcomes

Knowledge

After completing this module, the successful student will be able to:

- Explain in a discursive form the basic functions of management. (A6)
- Distinguish between different elements of organizational environment. (A6)
- Understand the rationale for the manager's decision making process. (A6)
- Appreciate the contribution of different management schools of thought to the science of management. (A6)
- Identify new trends in different management functions. (A6)

Skills

After completing this module, the successful student will be able to:

- Apply SWOT analysis to an organization. (D2, D4)
- Develop alternative solutions to specific managerial problems. (D2, D4)
- Present an analysis of a case study using appropriate tools. (D1)
- Practice working as a team to present research work. (D3)

Syllabus

- Basic managerial functions: planning, organizing, leading and controlling
- Different approaches to management: past and present.
- Managing in a global environment.
- The business environment
- Organization strategies through the use of SWOT analysis.
- The manager as a decision maker

Learning, Teaching and Assessment Strategies

Lectures will be used to introduce students to the main theoretical topics of the module. In-class discussions will be used to extend the scope of the lectures by encouraging students to explore the issues and ideas raised by the lecturer. Seminars will be used to enable students to apply management concepts to real world situations. Students are expected to carry out independent

study on a regular basis, as specified by the tutor. This might include further readings; it might also require work for specified written assignments. When students undertake project work, they will be supported by means of regular tutorials which will provide them with feedback on work in progress, and in-class presentations will involve the use of informal peer assessment.

Assessment Scheme

Written tests and unseen exams will be used to assess the students' understanding of the theoretical frameworks and their practical application.

A written report and its presentation will be used to assess the student's abilities to recognize various managerial practices.

- Written tests are used to assess students' understanding of core topics (10%) (outcomes 1-5)
- Written report (2000 words) and presentation based on a SWOT analysis (12% for report, 3% for presentation). Students will be provided with written and oral feedback on first drafts of reports. (outcomes 6,9)
- Brief written assignments based on case studies (10%).(outcomes 7, 8)
- Class Participation (5%). (outcomes 6,7, & 8)
- An unseen mid-term exam (20%) of 90 minutes and an unseen final exam (40%) of 3 hours will require students to answer questions (Multiple choice, and essays answers) on core theoretical issues. (outcomes 1-5)

Assessment Weighting

Coursework	40%
Exams	60%

Learning Materials

Essential

Robbins, Stephen, P. & Coulter, M. (2007). Management. New Jersey: Prentice Hall.

Recommended

Cole, G.A. (2005). Management Theory and Practice. Boston:Thomson.

Daft, Richard L. (2006). The New Era of Management. USA: Thomson/ South-western.

Schermerhorn, John R. (2006). Management. N.J.: John Wiley & Sons.

University Web-site, including: seminar activities, review questions, lecture notes and slides.