`

Philadelphia University

Faculty of Information Technology

Department of Information and Cyber Security

Synopsis of Modules

DESCRIPTION OF MODULES

0790110: Information Security Fundamentals

It establishes the core knowledge required of any cybersecurity role, the students will learn the foundations of threat, attacks & vulnerabilities, detect various types of compromise and have an understanding of penetration testing and vulnerability scanning concepts, compare and contrast Information Security roles, explain threat actor types, compare and contrast social engineering attack types topic, determine Malware Types, Assessing Security Posture with Software Tools, Explain Penetration Testing Concepts

Learning Outcomes:
On completion of this module, a student should have :
A1-A2-A4,B4,D1-D2-D4-D5

……..

0731110: Introduction to Information Systems and Technology

3 hours per week, 3 credit hours, prerequisite: None

Course (module) description:
This course introduces information systems and information technology, information systems development concepts, and application software. It identifies the basic types of business information systems, the major steps of the systems development process, and some of the strategies employed to lower costs or improve service. It explains how information is used in organizations and how I.T. improves quality, timeliness, and competitive advantage. It also defines the competitive advantages, types of roles, functions, and careers available in I.S.

Course (module) objectives:
Students should be acquainted with handling and managing data and information in business organizations and to understand the meaning of "Information Systems and technology and their effects on organizations and the different types of business information systems and the development life cycle.
Students must learn about different Computer Hardware and Software and different types of computer networks. Students should know how to deal with e-commerce

Course/ module components:
1.Books (title , author (s), publisher, year of publication)
Information Systems Essentials, Editors: Stephen Haag, Maeve Cumming; Published: McGraw-Hill/Irwin, Inc, 2009, Third edition.
- **Support material (s):** slides
- **Study guide (s) (if applicable).**
- **Homework and laboratory guide (s) if (applicable).**

Learning outcomes:
- Knowledge and understanding

1. Know and understand a wide range of principles and fundamentals of Information Systems and Information Technology.
2. The application of I.S. and I.T.

- Cognitive skills (thinking and analysis).

Basic analytical steps of Information Systems and defining the specifications of the I.T. required in business contexts**.**

• Communication skills (personal and academic):
1. Plan and undertake a small individual project in I.S. and I.T. fields.
2. Use the scientific literature effectively and make discriminating use of Web resources.
3. Present seminars in I.S. and I.T. fields.

• Practical and subject-specific skills (Transferable Skills):
1. Use appropriate computer-based tools.
2. Work effectively with and for others.
3. Strike a balance between self-reliance and seeking help when necessary in new situations.
4. Get knowledge about self-learning on the long run.

Assessment instruments:
• Short reports, presentations, Short research projects, Quizzes, and/or Home works (20%)
• First exam (20%)
• The second exam (20%)
• Final exam (40%)

……..

750113 Programming Fundamentals (1)

Course Hours: 3 hours per week, 3 credit hours (total of 48 hours)

Level: 1

Prerequisite: **None**

Aims:
This module introduces computer programming and emphasizes problem-solving on structured design fundamentals using the Top-Down problem-solving strategy (divide and conquer). This includes development, testing, implementation, documentation.
The module also explores programming logic via the algorithm concepts and implements them in programming structures, including functions, arrays, strings, and pointers.
.
Teaching Methods *Duration*: 16 weeks, 80 hours in total
Lectures: 32 hours (2 hours per week), *Tutorials*: 16 hours (1 per week),
Laboratories: 32 hours, 2 per week

Synopsis: problem-solving strategies, algorithmic language to describe such problem solving, introduces the principles of procedural programming, data types, control structures, data structures and functions, data representation on the machine level. Various problems are considered to be solved using C-like procedural programming language.

Assessment: Two 1-hour midterm exams (15% each); lab (30%); one 2-hours Final Examination (40%)

Textbook:

- P. Deitel & H. Deitel, **C++ How to program,** Pearson Education Limited, 2013.
- Guttag, John. **Introduction to Computation and Programming Using Python**. Spring 2013 edition. MIT Press, 2013. ISBN: 9780262519632. - MIT

……..

750114 Programming Fundamentals (2)

Course Hours: 3 hours per week, 3 credit hours (total of 48 hours)

Course module description

This course presents the fundamental concepts of programming using Python. It covers the basic structures of the programming tools such as variable names; data types; control structures; arrays; functions; Sequences (Strings, Tuples, Lists); Iteration; Dictionaries; Set; Modules; Exceptions; introduction to file processing; and Introduction to Object-oriented

Course module objectives

The objectives of this course are to:

1. Demonstrate an understanding of basic programming concepts, including data types, variables, modularity, parameters, conditional statements, iteration, and arrays.
2. Understand and manipulate several core data structures: Lists, Dictionaries, Tuples, and Strings.
3. Understand and employ objects, functions, and modularity.
4. Demonstrate methods of error handling
5. To do input/output with files in Python.
6. Demonstrate an understanding of basic and some advanced issues related to writing classes and methods

Course/ module components

- Books (title, author (s), publisher, year of publication)

Romano, Fabrizio (Author)
Publication Data: Birmingham: Mumbai Packt Publishing, 2018
ISBN: 978-1-78899-666-2
Learn Python programming: the no-nonsense, beginner's guide to programming, data science, and web development with Python 3.7

Teaching methods:

Duration: 16 weeks, 80 hours in total
Lectures: 32 hours (2 hours per week),
Tutorials: 16 hours (1 per week),
Laboratories: 32 hours, 2 per week

Student Learning Outcomes (SLO)

At the end of the course, students should be able to

- Knowledge and Understanding

A2. Know & understand a wide range of principles and tools available to the software developer, such as design methodologies, choice of algorithm, language, software libraries, and user interface technique:
A4. Know & understand a wide range of software and hardware used in the development of computer systems
A5. Know & understand the professional and ethical responsibilities of the practicing computer professional, including understanding the need for quality, security, and computer ethics.

- Intellectual Skills

B1. Analyze a wide range of problems and provide solutions through suitable algorithms, structures, diagrams, and other appropriate methods

B4. Practice self learning by using the e-courses

- Professional and Practical Skills

C3. Work effectively with and for others.
C4. Strike a balance between self-reliance and seeking help when necessary in new situations
C5. Display personal responsibility by working to multiple deadlines in complex activities

- General and Transferable Skills

D2. Prepare and deliver coherent and structured verbal and written technical reports.
D4. Use the scientific literature effectively and make discriminating use of Web resources
D5. Design, write and debug computer programs in appropriate languages

Learning outcomes achievement

- Development: A2, A4, and A5 are developed through lectures and laboratory sessions.
  B1, D5, C3, and C4 have developed through Tutorials and Lab sessions,
  B4, D2, D4, D5, and C5 are developed through Homework

- Assessment: A2, A4, A5, B1, D5, and C4 and are assessed through Quizzes, written exams, and Practical Works Exams.
  B4, D2, D4, D5, and C5 are assessed through the Homework Exam.

……..

0750120 Discrete Mathematics

Level: 1

Prerequisite: **None**

Aims:
This course introduces Discrete Mathematics for students from the I.T. majors, covering main topics in number theory, propositional logic, proof techniques, sets and relations, counting techniques, graph theory, and selected applications in computer algorithms.

Teaching Methods: 38 hours Lectures (2-3 per week) + 10 hours Tutorial

Synopsis:

Logic: logic operators AND, OR, IFF, XOR, truth table, tautology, equivalence. Standard forms, predicates, and quantifiers. Sets: set operations, set identities, power set, cardinality, cross product, power set. Modulo operation, divisibility, GCD and LCM, the Euclidean algorithm. Combinatorics: the addition and multiplication principles, the Pigeonhole principle. Inclusion-exclusion principles, permutations and combinations, permutations on multisets. Recurrence relation: solving first-order homogeneous sequences. Methods of proof: mathematical induction. Relations: properties of relations, representation by digraphs, zero-one matrices, transitive closures. Equivalence relations, partial order relations, total order, Hasse diagrams. Graph Theory: complete graphs, complete bipartite, representations by an adjacency matrix, incidence matrix, distance matrix. Trees, minimal spanning trees, Euler circuit, the Chinese postman problem. Coloring algorithms, planar graphs, maps, and dual graphs.

Modes of Assessment:

Two 1-hour midterm exams (20% each); Assignments (20%); One 2-hours Final Examination (40%)

Textbook
Amin Witno, Discrete Structures in Five Chapters, CreateSpace 2010
……

3. Intermediate Modules

0790211 : Principles of Cryptography
The students will learn the foundations of basic cryptography concepts, compare and contrast basic cryptography concepts; what is Cryptography? And the history of Cryptography, Mono-Alphabet Substitution, Multi-Alphabet Substitution, Homophonic Substitution, Null Ciphers Book Ciphers, Rail Fence Ciphers, the Enigma Machine, CrypTool, explain Hashing and Symmetric cryptographic algorithms, explain Asymmetric cryptographic algorithms, implementing a Public Key Infrastructure, implement Certificates and Certificate Authorities, implement PKI Management.

Learning Outcomes:
On completion of this module, a student should have:
A1-A2-A4,B4, C1-C2-C3-C9, D1-D2
……

0750250, Computing ethics and technical writing

Providing Department: Software Engineering, Faculty of I.T.
***Module Coordinator(s)*:**
Year: 2
Credit: 3 credit hours
Prerequisite: 731110

Aims:
This module aims to give students an informed awareness of the principal issues of professional ethics and responsibility (ergonomics and ethics) in the analysis, design, implementation, and use of computers, information systems, and Information Technology (I.T.) products. This will help students in recognition of ethical problems when they occur. Also, it will enable students to deal effectively with ethical, social, and professional issues now and in their future careers.

Teaching Methods: 36 hours Lectures (2-3 per week) + 9 hours Projects (classwork) (average 1 per week) + **3 hours Seminars (1 per month)**

Learning Outcomes:
On completing this module, students will:
1. Understand the basic concepts of ethics, morals, law, ergonomics, and profession.
2. Be aware of the requirements for accreditation in respect of Professional Issues.
3. Have a basic knowledge of Intellectual Property Rights (IPR) concerning Copyright and Patents.
4. Be aware of some of the potential problems of managing large I.T. projects by professional and ethical issues.
5. Be aware of the requirements for professionalism regarding the work of the professional societies and their codes of conduct and practice.
6. Have acquired basic knowledge of the Data Protection Act and its implications.
7. Be able to assess and evaluate the legal aspect of workplace practices.
8. Be able to asses and evaluate the impacts of I.T. technology on society and culture.
9. Be aware of Jordanian Professional Issues.

Assessment of Learning Outcomes:

Learning outcomes are assessed through examination and individual and group case studies, which require demonstration of the use of a combination of the learning outcomes to be employed in producing the essays and presentations.

Contribution to Programme Learning Outcomes:
A2, A5, B1, C1, C3, C4, C6, D3, D4.

Synopsis: Introduction to Ethics; Professional and Professionalism; Code of Ethics and Social Issues; Computer/I.T. professionals; Computer Security; Privacy and Internet Issues; Information Systems and Ethics; Associations of I.T. professionals; Ethics and the Internet; Ethical Challenges of e-Business; Ethical Challenges of e-Business; Continuous Professional Development; Intellectual Property Rights; Jordanian Codes for Intellectual Property Rights; Seminars and Project Discussion.

Modes of Assessment:
Two 1-hour midterm exams (20% each); Assignments (20%); One 2-hours Final Examination (40%)

Textbooks and Supporting Material:
1. Deborah G. Johnson, Computer Ethics. 3ed Edition, Englewood Cliffs, N.J., Prentice-Hall, 2001.
2. Gorge Reynoids, Ethics in Information Technology, Thomason, 2003.
3. Sara Baase, A Gift of Fire: Social, Legal and Ethical Issues for Computer and the Internet, 2nd ed., 2003.
4. Tavani H. T. and Hoboken N. J., Ethics and Technology, John Wiley, 3rd ed, 2004.
5. مجموعة تشريعات الملكية الفكرية الأردنية

Website(s):
ACM, IEEE and BCS Web Sites.
www.cyberethics.cbi.mssttste.edu
www.aitp.org
www.acm.org
www.prenhall.com
www.jcs.rg.jo

……..

721223, Object-Oriented Programming

Providing Department: Software Engineering, Faculty of I.T.
***Module Coordinator(s)*:**
Year: 1
Credit: 3 credit hours
Prerequisite: 750114

Aims:
This module introduces the concepts of object-oriented programming for students after having a background in the procedural paradigm. It aims to develop an understanding of the principles of the object-oriented paradigm, provide familiarity with approaches to object-oriented modeling and design, provide a familiarity with the syntax, class hierarchy, environment, and simple application construction for an object-oriented programming language. The module emphasizes modern software engineering principles and developing fundamental programming skills in a language that supports the object-oriented paradigm (Java, for instance).

Teaching Methods: 32 hours Lectures (2 per week) + 16 hours Tutorial (1 per week) + 16 hours Laboratory (1 per week)

Learning Outcomes:
A student completing this module should:
1. Acquire a whole Object-Oriented Thinking (A)
2. Have a clear understanding of object-oriented concepts such as objects, classes, inheritance, and polymorphism. (A)
3. Have an informal understanding of object-oriented programs' operational semantics in terms of creating objects and messages passing between them. (A)
4. Be able to design small Object-oriented programs which meet requirements expressed in English, with a strong software engineering foundation (B)
5. Know Object-Oriented Design guidelines. (A, B)
6. Be able to code small software systems in Java language. (C).
7. Be able to maintain large, high-quality software systems (C)

Assessment of Learning Outcomes:
Learning outcomes (1), (6), and (7) are assessed by examination and laboratory. Learning outcomes (2), (3), and (7) are assessed by tutorial and examination. Learning outcomes (4) and (5) are assessed in the laboratory.

Contribution to Programme Learning Outcomes:

A2, A3, B1, B2, C5, D2, D4

Synopsis: Introduction to Object-Oriented Thinking: Object Modeling; Objects and Classes; Understanding Class Definition; Object Interaction (1): Overloading; Object Interaction (2): Composition; Grouping Objects; Using Library Classes; More Sophisticated Behavior: Information Hiding; Inheritance (1): Reuse, Inheritance (2): Sub-typing; Inheritance (3): Polymorphism, Overriding; Abstract Classes, Abstract Methods, Interfaces, Multiple inheritances; Exception Handling; Designing Applications

Modes of Assessment:
Two 1-hour midterm exams (20% each); Assignments (20%); One 2-hours Final Examination (40%)

Textbooks and Supporting Material:
1- David j. Barnes And Michael Kolling, Objects First with Java: A Practical Introduction using BlueJ, Prentice Hall, Pearson Education, 2nd Edition, 2005

……..

721224, Data Structures

Providing Department: Software Engineering, Faculty of I.T.
***Module Coordinator(s)*:**
Year: **2**
Credit: 3 credit hours
Prerequisite: 721223+250104

Aims:
This is a **programming-intensive** module where students learn the fundamentals of designing data structures in complex programs. The data structures course is essential for computer scientists and anyone who will ever undertake any serious programming task. This course deals with the fundamentals of organizing and manipulating data efficiently using clean conceptual models. Students study many important conceptual data types, their realization through implementation and analysis of their efficiency. This course's implementation is carried out in the Java programming language, but the principles are more generally applicable to most modern programming environments.
Topics include recursion, the underlying philosophy of object-oriented programming, fundamental data structures (including stacks, queues, linked lists, hash tables, trees, and graphs), and the basics of algorithmic analysis.

Teaching Methods: 32 hours Lectures (2 per week) + 16 hours Tutorial (1 per week)

Learning Outcomes:
On successful completion of this module, student will:

1. build on understanding of basic ideas about data structures given in the prerequisite module (A)
2. understand the basic concepts of time and space complexity (A)
3. be able to manipulate recursive algorithms (B)
4. be able to develop efficient algorithms for manipulating data structures (B)
5. know a range of algorithm structures and how to implement them (A, B, C)
6. know and understand a wide range of searching and sorting algorithms (A, B)
7. understand how the Abstract Data Type (ADT) is used (A)
8. understand several representations of trees, and their applications (A, C)
9. understand several representations of graphs, and their applications, together with a selection of important algorithms on graphs (A, C)
10. be able to construct and use the data structures mentioned above. (A, B, C)

Assessment of Learning Outcomes:
Outcomes 1 to 10 are assessed by coursework and examinations
Contribution to Programme Learning Outcomes:
A1, B1, B2, C5, D6

Synopsis: Introduction to Software Engineering, Introduction to data structures: data structures and algorithms; Data Design and Implementation; Algorithm complexity; List ADT: static implementation, singly linked list; List ADT: dynamic implementation, singly linked list; Lists: doubly linked list and circular linked list; Stacks: Static implementation and dynamic implementation; Queues: Static implementation and dynamic implementation, circular queue; Programming with Recursion; Trees: Binary search tree; Trees: binary expression tree, and heap tree; Priority Queues and Heaps; Graph ADT; Sorting: Bubble sort, selection sort, insertion sort, Quicksort, Heap sort; Searching: Sequential search, Binary Search; Hashing: hash function, Separate chaining, open addressing

Modes of Assessment:
Two 1-hour midterm exams (15% each); Coursework (15%); Tutorial Contribution (5%); Final (unseen) Exam (50%)

Textbooks and Supporting Material:
1- Nell Dale, Daniel T. Joyce and Chip Weems, Object-Oriented Data Structures using Java, Jones and Bartlett Publishers, 2001
2- Goodrich and Tamassia, Data Structures and Algorithms in Java, 2nd edition, John Wiley and Sons, 2000, ISBN 0471383678.
3- Arnold, Gosling, and Holmes, The Java Programming Language, 3rd edition, Addison-Wesley, 2000, ISBN 0201704331.
4- Mark Allen Weiss, Data Structures and Algorithm Analysis in C++, Addison-Wesley, 1999

……..

731221, Database Fundamentals

Providing Department: Computer Information Systems, Faculty of I.T.
Module Coordinator(s):
Year: 2
Credit: 3 credit hours
Prerequisite: 721223

Aims:
This module aims to give the students the main concepts of database, design the database, database models, normalization techniques, query languages, Object-Oriented database, query optimization and database and the web. Further, the students have to practice and write some applications regarding the database.

Teaching Methods: **26 hours Lectures (average 2 per week) + 16 hours Laboratory (1 per week) + 6 hours Tutorials (1 each fortnight)**

Learning Outcomes:

When completing this module, a student should be able to:

1. Discuss/explain the importance of data and the difference between file management and databases. (A)
2. Discuss/explain the design of database management system architectures and environments. (A)
3. Discuss/explain the principles of database design. (A)
4. Discuss, explain and apply conceptual design methodologies, in particular, conceptual design using Extended Entity-Relationship modeling. (A, B, C, D)
5. Discuss, explain and apply the relational model and mappings from conceptual designs, in particular normalizations. (A, B, C, D)
6. Discuss/explain physical and performance-related design considerations. (A)
7. Discuss/explain transaction processing. (A)
8. Discuss, explain and apply SQL and the Oracle DBMS. (A, C, D)

Assessment of Learning Outcomes:

Examinations assess learning outcomes (1) through (7). Learning outcomes (3), (4), and (8) are assessed by projects design and implementation.

Contribution to Programme Learning Outcomes:

A2, A3, A4, A5, B1, B2, B3, C1, C2, C6, D1, D3

Synopsis: Introduction to Database and DBMS; Database Models; Database Design; Relational Algebra and Relational Calculus; Query Languages (SQL); D.B. normalization; Database Integrity and Security; Indexing Techniques; Query Optimization; Distributed Data Base; Object-Oriented Database

Modes of Assessment:

Two 1-hour midterm exams (20% each); Assignments (20%); One 2-hours Final Examination (40%)

Textbooks and Supporting Material:

1- A. Silberschatz, H.F. Korth & S. Sudarshan, Database System Concepts, McGraw Hill, 2002
2- El-Masri & Navathe, Fundamentals of Database Systems, Prentice-Hall, 2002.
3- Jeffry Ullman, Principles of Database Systems, S.U. Publishers, 1999
4- C.J. Date, An Introduction to Database Systems, Addison Wesley, 1995

…….

731340, Fundamentals of Computer Networks

Providing Department: Computer Information Systems, Faculty of I.T.
Module Coordinator(s):
Year: 3
Credit: 3 credit hours
Prerequisite: 721224

Aims:

This module is the first module of the curriculum related to the computer network field. Its aim is to provide students with broad coverage of the basic computer networking concepts of the four layers of ISO, circuit switch, packet switch, etc.

The module, however, does not focus on a detailed study or cover the technologies. The concepts given in this module will be sincerely handled in the next level module (750441).

Teaching Methods: 32 hours Lectures (2 per week (including two 1-hour midterm exams)) + 16 hours Tutorial (1 per week) + 16 hours Laboratory

Learning Outcomes:
A student completing this module should be able to:
1. Discuss necessary network standards in their historical context (A)
2. Describe the responsibilities of the first four layers of the ISO reference model. (A)
3. Discuss the differences between circuit switching and packet switching along with the advantages and disadvantages of each. (A, B)
4. Explain how a network can detect and correct transmission errors. (A, B)
5. Illustrate how a packet is routed over the Internet. (C)
6. Install a simple network with two clients and a single server using standard host-configuration software tools such as DHCP. (C, D)

Assessment of Learning Outcomes:
Learning outcomes (1) - (3) are assessed by examination and tutorials. Learning outcomes (4) – (6) are assessed by assignments and seminars.

Contribution to Programme Learning Outcomes
A3, A4, B2, C6, D2, D5, D6.
Synopsis: Introduction; Network Model; Data and Signal**;** Digital signal; Analog signal; Switching; Error Detection and Control; Error Detection and Control; Data Link Control; Multiple Access; Network Layer: Logical Addressing; Network Layer: Delivery, Forwarding, and Routing; Network Layer: Delivery, Forwarding, and Routing; Process-to process Delivery; Congestion Control and Quality of service.

Modes of Assessment:
Two 1-hour midterm exams (20% each); Assignments (20%); One 2-hours Final Examination (40%)

Textbooks and Supporting Material:
1- Behrouz A. Forouzan, Data Communications and Networking, McGraw Hill Higher Education, Fourth Edition, 2007
2- Andrew S. Tanenbaum, Computer Networks, Prentice Hall, Last Edition.
Website(s):
www.mhhe.com/forouzan

750215, Visual Programming

Providing Department: Computer Information Systems, Faculty of I.T.
Module Coordinator(s)**:**
Year: 2
Credit: 3 credit hours
Prerequisite: **721223**

Aims: This module aims to provide students capabilities to design and implement the applications using visual programming through Microsoft Visual Studio .Net and V.C. # to develop different types of applications using .Net platform.

Teaching Methods: 32 hours Lectures (2 per week) + 12 hours Tutorials (on average 1 per week) + 16 hours Laboratory (1 per week) + 4 hours Seminar

Learning Outcomes:

On completing this module you should:

1. Have a clear understanding of what comprises a correct program in C# through .Net frame components (A)
2. Have a clear understanding of the object-oriented terminology used to describe C# and V.C. # project features with their visual components. (A, C)
3. Have an informal understanding of object-oriented programs' operational semantics in creating objects and messages passing between different interfaces. (A)
4. Be able to design, code, and test C# project, which meets requirements expressed in English. (B, C, D)
5. Be able to understand the documentation for, and make use of, the MSDN library. (A, C)
6. Have a good understanding of the different focuses at various stages of the development process. (A, C, D)
7. Have knowledge of design GUI with visual components guidelines. (A, B)
8. Be able to apply the guidelines in learning outcome (7) to a real design problem and justify how they have been used. (A, B)
9. Be able to write a project in C# and V.C. #, which implements the design in learning outcome (8). (C).
10. Be able to work effectively alone or as a small group member working on some programming tasks. (C, D)

Assessment of Learning Outcomes:

Learning outcomes (1), (6), and (8) are assessed by examination and laboratory; learning outcomes (2), (3), and (7) are assessed by tutorial and examination; learning outcomes (4), (5), (9) and (10) are assessed in the laboratory.

Contribution to Programme Learning Outcomes:

A2, A3, A4, B3, C5, C6, D1, D2, D4, D5

Synopsis: **Introducing the Microsoft .NET Platform:** .NET Platform, .NET and Windows DNA, .NET Architecture Hierarchy, .NET Platform features, Multilanguage Development, Platform and Processor Independence, .NET Components, Common Type System CTS, Common Language Specification CLS , .NET Base Class Library (BCL); **Visual Studio.NET IDE:** Visual Studio.NET, Components of VS.NET, Design Window, Code Window, Server Explorer, Toolbox, Docking Windows, Properties Explorer, Solution Explorer, Object Browser, Dynamic Help, Task List Explorer, Features of VS.NET, XML Editor, Creating a Project, Add Reference, Build the Project, Debugging a Project; **Introducing C# Programming:** Data Types, Value Types, Reference Types, Control Structures (if, if-else, switch, for, while, do while, break, continue, return, goto), Understanding Properties and Indexers Accessing Lists (Array) with Indexers, Events, Exception Handling, Using OOP (Object, Class, Constructor/destructor, Inheritance, Polymorphism, Encapsulation); **Windows Forms:** Windows Forms, Adding Controls, Adding an Event Handler, Adding Controls at Runtime, Attaching an Event Handler at Runtime, Writing a Simple Text Editor, Creating a Menu, Adding a New Form, Creating a Multiple Document Interface, Creating a Dialog Form, Using Form Inheritance, Adding a *TabControl*,, Anchoring Controls, Changing the Startup Form, Connecting the Dialog, Using the ListView and TreeView, Controls, Building an *ImageList*, Adding a *ListView*, Using the Details View, Attaching a Context Menu, Adding a *TreeView*, Implementing Drag and Drop, Creating Controls, Creating a User Control, Adding a Property, Adding Functionality, Writing a Custom Control, Testing the Control, Enhancing the Control, Sub classing Controls; **Graphics and Multimedia:** Graphics Contexts and Graphics Objects, Color Control, Font Control, Drawing Lines, Rectangles and Ovals, Drawing Arcs, Drawing Polygons and Polylines, Advanced Graphics Capabilities, Introduction to Multimedia, Loading Displaying and Scaling Images, Animating a Series of Images, Windows Media Player, Microsoft Agent; **ADO.NET:** ADO.NET Architecture, Understanding the *Connection*Object, Building the *Connection* String, Understanding the *Command*Object, Understanding *DataReaders*, Understanding *DataSets* and *DataAdapters*, DataTable, *DataColumn*, *DataRow*, Differences between *DataReader* Model and *DataSet* Model, Understanding the *DataView*Object ,Working with System.Data.OleDb, Using *DataReaders*, Using *DataSets*, Working with SQL.NET, Using Stored Procedures, Working with Odbc.NET, Using DSN Connection; **Multithreading:** Thread States: Life Cycle of a Thread, Thread Priorities and Thread Scheduling, Thread Synchronization and Class Monitor, Producer/Consumer Relationship without Thread,

Synchronization, Producer/Consumer Relationship with Thread Synchronization, Producer/Consumer Relationship: Circular Buffer; **Networking:** Introduction, Establishing a Simple Server (Using Stream Sockets), Establishing a Simple Client (Using Stream Sockets), Client/Server Interaction with Stream-Socket Connections, Connectionless Client/Server Interaction with Datagrams, one Server multi-Clients system; **ASP.NET:** Introducing the ASP.NET Architecture, ASP.NET Server Controls, Working with User, Controls, Custom Controls, Understanding the Web.config File, Using the Global.asax Page,

Modes of Assessment: Two 1-hour midterm exams (20% each); Assignments (20%); One 2-hours Final Examination (40%)
)

Textbooks and reference books:

1- H. M. Deitel & J. Deitel, "C# How to Program", Prentice Hall, 2014 fifth edition
2- A.Turtschi et.al. " Mastering Visual C# .Net", Sybex 2002
3- Eric Gunnerson, "A Programmer's Introduction to C#", Apress 2000
4- Anders Hejlsberg et.al. "C# Language Reference", Microsoft Corporation 2000
5- Erric Buttow et al. "C#, your visual blueprint for building .Net application", Hungry Minds 2002
6- Charles Carroll, "Programming C#", O'Reily & Associates 2000
Karh Watson "Beginning C#" Wrox Press 2001.

……

0731213 Introduction to Web Programing

3 hours per week, 3 credit hours, prerequisite: 0750114

Course description

This course is intended to give the student advanced issues in website design and implementation. At the course completion, students will have the know-how of designing and implementing web-based applications, completely database-driven websites.

The course involves two main parts:

- Advanced client-side programming.
- Advanced server-side programming.

Course objective

On completing the module, the students are expected to have gained a good knowledge of:

- Implementing advanced server-side programming
- Improving personal productivity concepts through web authoring.

Course components:

- Books (title, author (s), publisher, year of publication)
Gosselin, don, PHP Programming with MySQL. Course Technology Incorporated, 2005, ISBN 0-619-21687-5
- Support material (s)
The world's largest web development site: www.w3schools.com
- Study guide (s) (if applicable).
- Homework and laboratory guide (s) if (applicable).

Teaching methods

Lectures, discussion, groups, tutorials, problem solving, etc.

Learning outcomes:

- Knowledge and understanding.
- Cognitive skills (thinking and analysis).
- Communication skills (personal and academic).
- Practical and subject-specific skills (Transferable Skills).

Assessment instruments:

- Short reports, Assignments, Projects, Quizzes, and/or Home works (20%)
- The First exam (20%)
- The second exam (20%)
- The Final exam (40%)

750323, Algorithms

Providing Department: Information and Cyber Security, Faculty of I.T.
Module Coordinator(s):
Year: 3
Credit: 3 credit hours
Prerequisite: 750272 + 721223

Aims:
This module aims to learn how to develop efficient algorithms for simple computational tasks and reasoning about their correctness. Through the complexity measures, a different range of behaviors of algorithms and the notion of tractable and intractable problems will be understood. Furthermore, the module introduces formal techniques to support the design and analysis of algorithms, focusing on both the underlying mathematical theory and practical considerations of efficiency. Topics include asymptotic complexity bounds, techniques of analysis, and algorithmic strategies.

Teaching Methods: 38 hours Lectures (2 per week (including two 1-hour midterm exams)) + 10 hours Tutorials (average 1 hour per week)

Learning Outcomes:

When completing this module, you should be able to:

1. understand basic ideas about algorithms (A)
2. develop efficient algorithms for simple computational tasks (B)
3. reason about the correctness of algorithms (B)
4. understand the concepts of time and space complexity, worst case, average case and best case complexities, and the big-O notation (A)
5. compute complexity measures of algorithms, including recursive algorithms using recurrence relations (B)
6. understand the range of behaviors of algorithms and the notion of tractable and intractable problems (A, B)
7. know and understand a wide range of searching and sorting algorithms (A, B)

Assessment of Learning Outcomes:
Examinations and tutorials assess all learning outcomes. In addition, learning outcomes (4), (5), and (6) are assessed by examinations and coursework.

Contribution to Programme Learning Outcomes:
A1, A2, B1, B2, B3

Synopsis: Introduction, Algorithm definition, Algorithm Analysis; **Mathematical Induction;** Summation Techniques; Recurrence Relations; **Design & Analysis of Algorithms: Divide and conquer**, Greedy Algorithm, Dynamic Programming, Backtracking, Branch-Bound; Lower Bound Theory; Sorting and Searching; NP-Complete Problems: Basic Concepts, NP-Hard & NP-Complete Problem

Modes of Assessment:
Two 1-hour midterm exams (15% each); Tutorial contributions (5%), Coursework (15%); Final written Examination (50%)

Textbooks and Supporting Material:
1- Jon Kleinberg, Eva Tardos, Algorithm design**,** Boston: Pearson Education Limited, 2014.
2- Alwan, Raad F., Design and Analysis of Algorithms, Dar Majdalawi Publication & Distribution, Amman, 2010.
3- Sara Baase, Computer Algorithms: Introduction to Design and Analysis, Third Edition, Addison-Wesley, 2000.
4- Udi Manber, Introduction to Algorithms: a Creative Approach, Addison-Wesley, 1997.
5- T. Cormen, et.al., Introduction to Algorithms, 1999.
6- R. Sedgewick, Algorithms in C++, 2002.

731321, Systems Analysis and Design

Providing Department: Management Information Systems, Faculty of I.T.
Module Coordinator(s):
Year: 3
Credit: 3 credit hours
Prerequisite: 721111
Aims:

This module introduces the students to the concepts and skills of system analysis and design. It includes expanded coverage of data flow diagrams, data dictionaries, and process specifications, as it introduces examples of new software used by analysts, designers to manage projects, analyze and document systems, design new systems and implement their plans. It also introduces a recent coverage of UML, wireless technologies, and ERP; web-based systems for e-commerce and expanded coverage on RAD and GUI design.

Teaching Methods: **32 hours Lectures (2 per week) + 8 hours Tutorials (1 per fortnight) + 8 hours Seminars (in the last 3 weeks)**

Learning Outcomes:
At the end of this module, the student will be able to:
1- Understand the principles and tools of systems analysis and Design (A).
2- Solve a wide range of problems related to analyzing, designing, and constructing information systems (A, B, C).
3- Understand the application of computing in different contexts (A).
4- Understand the professional and ethical responsibilities of practicing the computer professional, including understanding the need for quality. (A)
5- Plan and undertake a major individual project, prepare and deliver coherent and structured verbal and written technical reports (B, C, D).
6- Analysis and design of systems of small sizes. (B, C)

Assessment of Learning Outcomes
Learning outcomes (1) – (4) are assessed by examinations, tutorial and in the assignments. Learning outcomes (5) and (6) are assessed by seminars and projects

Contribution to Programme Learning Outcomes:
A2, A3, A4, A5, B1, B3, C1, C2, C5, D2, D4, D5.

Synopsis: Systems Analysis Fundamentals: Introducing Systems Analysis and Design Concepts, roles of systems analysts, system development life cycle, using CASE Tools, depicting system graphically, determine feasibility, activity planning, and control; Information requirements analysis: Sampling and investigating data, interviewing, using questionnaires, prototyping; The analysis process: Using data flow diagram, using data dictionaries, describing process specifications and structured decisions, the system proposal; The essential of Design: Designing output, designing the files or database, designing the user interface, designing data entry forms, documenting the design phase; Software engineering and implementation: Quality assurance through software engineering, implementing the information system, Object-Oriented analysis and design; different software tools will be used in this course.

Modes of Assessment:
Two 1-hour midterm exams (15% each); Assignments (15%); Tutorial Contribution (5%); Final Examination: written (unseen) exam (40%) + final project (10%)

Textbook and Supporting Material:
1. Kenneth E. Kendall and Julie E. Kendall, Systems Analysis and Design 5th Edition, Prentice Hall PTR, 2001
2- Silver and Silver, System Analysis and Design, Addison Wesley, Last Edition

3. Advanced Modules

0790325: Forensic and Investigations

Digital or computer forensics focuses on the digital domain, including computer forensics; the students will learn the following topics: Computer Forensics in Today's World, Computer Forensics Investigation Process, Understanding Hard Disks and File Systems, Operating System Forensics, Defeating Anti-Forensics Techniques, Data Acquisition and Duplication, Network Forensics, Investigating Web Attacks, Database Forensics, Cloud Forensics, Malware Forensics, Investigating Email Crimes, Mobile Forensics, Investigative Reports.

Learning Outcomes:
On completion of this module, a student should have:
A2-A3-A4-5, B1-B2-B4-B5-B6, C1-C2-C3-C4-C5-C6-C7-C8-C9,D1-D2-D3-D4-D5-D7

……

0790322: Wireless Computer Network

This course provides the networking professional a complete foundation of knowledge for entering into or advancing in the wireless networking industry. From basic R.F. theory to 802.11 frame exchange processes, this course delivers hands-on training that will benefit the novice as well as the experienced network professional. This intensive course covers all that is required to prepare for the CWNA Certification.

Learning Outcomes:
On completion of this module, a student should have:
A2-A4-A5, B1-B2-B6, C1-C2-C3-C4-C6-C9,D2-D3-D4-D5-D7

……

0790323: Linux Operating Systems

This course provides the Linux Operating Systems professional a complete foundation of knowledge for entering into or advancing in the Hardware & System Configuration Configure kernel modules, network parameters, storage, cloud and virtualization technologies, System Operation & Maintenance Manage software and services, and explain server roles, job scheduling, and the use and operation of Linux devices, Security: Understand best practices for permissions and authentication, firewalls, and file management. Linux Troubleshooting & Diagnostics: analyze system properties and processes and troubleshoot user, application, and hardware issues, Automation & Scripting: Execute basic BASH scripts, version control using Git, and orchestration processes.

Learning Outcomes:
On completion of this module, a student should have:
A2-A3-A4-A5, B2-B3-B4-B6, C1—C2-C3-C4-C7-C8, D4-D5-D7

……

0790422: Networks Security

This course focuses on creating network administrators who are trained to protect, detect, and respond to the network's threats. Network administrators are usually familiar with network components, traffic, performance and utilization, network topology, location of each system, security policy, etc. The students will get a fundamental understanding of the valid construct of data transfer, network technologies, and software technologies to understand how networks operate, what software is automating, and how to analyze the subject material. In addition, network defense fundamentals, the application of network security controls, protocols, perimeter appliances, secure IDS, VPN and firewall configuration, intricacies of network traffic signature, analysis and vulnerability scanning are also covered, which will help the Network Administrator design greater network security policies and successful incident response plans.

Learning Outcomes:
On completion of this module, a student should have :
A2-A3-A4-5, B1-B2-B4-B6, C1-C2-C3-C4-C6-C7-C8-C9,D2-D3-D4-D5-D7

……

0790421: Networks Surveillance and Documentations

The students will learn the basics of Threat Intelligence, Threat Intelligence Lifecycle and Frameworks, Cyber Threats and Kill Chain Methodology, Understanding Advanced Persistent Threats (APTs), Understanding Indicators of Compromise (IoCs), Requirements, Planning, Direction, and Review, Understanding Organization's Current Threat Landscape, Understanding Requirements Analysis, Planning Threat Intelligence Program, Establishing Management Support, Data Analysis, Understanding Data Analysis Techniques, Understanding Threat Analysis Process, Overview of Fine-Tuning Threat Analysis.

Learning Outcomes:

On completion of this module, a student should have :
A2-A3-A4-5, B1-B2-B4-B6, C1-C2-C3-C4-C6-C7-C8-C9,D2-D3-D4-D5-D7

……

0790423: Information Security Risks Management

This course delivers expertly curated information that enables students to master the risk management process related to all assets of relevance for Information Security using the ISO/IEC 27005 standard as a reference framework. Taking this course will enable students to gain a thorough understanding of best practices of risk assessment. In addition, the course prepares students to support organizations, prioritize risks and undertake appropriate actions to reduce and mitigate them.

Learning Outcomes:
On completion of this module, a student should have :
A2-A3-A4-5, B1-B2-B4-B6, C1-C2-C3-C4-C6-C7-C8-C9,D2-D3-D4-D5-D7

……

0790440 , Research Project (1) + 0790440 , Research Project (2)

Providing Department: Information and Cyber Security, Faculty of I.T.
Module Coordinator(s):
Year: 4
Credit: 3 credit hours
Prerequisite: 750398

General Descriptions:
The graduation project consists of a single project on which the student works over 16 weeks that can be extended to 32 weeks (2 semesters). It is assumed that the student spends a nominal 192 hours (or 384 hours), the equivalent of 12 hours per week, working on this. There are three deliverables: demonstration, discussion, and a written report.

A student works under the supervision of a member of staff, the Supervisor. Most of the projects involve three students working together on the same project; apart from these, all students do different projects.

- How to choose a project
- Organization for projects
- Demonstrations
- Report Standards
- Staff List

Aims:
The aims for the project work done in the fourth year are:
1- To manage and execute a substantial project in a limited time.
2- To identify and learn whatever new skills are needed to complete the project.
3- To apply design and engineering skills in the accomplishment of a single task. In this context the skills mentioned may be in the general area of design and engineering in its broadest sense, or may be very specifically related to particular tools.

Teaching methods: Duration: 32 weeks (2 semesters) starts in first semester: Lectures: 6 or 7 in total, spread through the 2 semesters + Laboratories: none scheduled, 120 hours expected through semester

Learning Outcomes:
On completion of this module, a student should have

1. Used the project supervisor appropriately as project consultant or customer. (D)
2. Planned, executed and completed a significant design and, as appropriate, implementation within the time budget available. (B, C)
3. Given a demonstration showing practical competence and demonstrating the results of the project. (C).
4. Documented the project in a final report. (C)

Assessment of Learning Outcomes:
There is no examination.
The Supervisor assesses learning outcome (1). Learning outcomes (3) is assessed by the project examination committee and by judging the demonstration. Learning outcome (4) is assessed by judging the report. All of these mechanisms assess learning outcomes (2).

Modes of Assessment:
Supervisor mark: 35% + Project Examination Committee mark: 65% (demonstration 20%, Report 25%, discussion 20%)

Contribution to Programme Learning Outcomes
B1, B2, B3, C1, C2, C3, D1, D2, D3, D4, D5

Syllabus
The occasional lectures are on topics of particular interest to students doing a project in their final year.

- Overview of projects and project assessment.
- Career advice.
- How to give a seminar.
- Writing English.
- How to give a demonstration.
- How to write a project report.