1. Course Information

Faculty	Science
Department	Computer Science
Code	CSI131
Title	Discrete Structures I
Credits	3
Type	Core
Semester in which course is taught (1 or 2)	1
Pre-requisites	none
Co-requisites	none

Aims and Learning Objectives

To introduce students to the mathematical tools of logic and induction, basic definitions and theorems concerning sets, basic counting, and algorithms.

Learning Outcomes

After completion of the course, students should be able to

- read and understand basic mathematical proofs
- state and prove non-trivial facts
- recite mathematical ideas and notation used in Computer Science
- describe the importance of algorithms in solving problems
- demonstrate and apply basic counting principles

Course Synopsis

Functions, relations and sets; Propositional logic and predicate calculus; Algorithms as effective processes; Proof techniques; Algorithms, the integers and matrices; Basic counting; Boolean algebra.

Course Delivery

2 hours lectures, 2 hours lab/tutorial

Modes of Assessment

CA and Final Examination

Reading (and other resources) List

1. Rosen K., "*Discrete Mathematics and Its Applications*", 5th Edition, McGraw-Hill, 2003.
2. Ensley D., and Crawley J., "*Discrete Mathematics: Mathematical Reasoning and Proof with Puzzles, Patterns, and Games* ", John Wiley & Sons, Inc., 2006.
3. Cupillari A., *"The Nuts and Bolts of Proofs"*, Academic Press, 2001.
4. E-resources.

Faculty	Science
Department	Computer Science
Code	CSI132
Title	Discrete Structures II
Credits Type	3 Core
Semester in which course is taught (1 or 2)	2
Pre-requisites	CSI131
Co-requisites	none

Aims and Learning Objectives

To further the student's knowledge of counting techniques. The course will introduce the notions of graphs and trees and their applications in problem solving. It will also introduce students to the theory of modelling computations.

Learning Outcomes

After completion of the course, students should be able to

- read and understand mathematical proofs
- state and prove non-trivial facts
- recite mathematical ideas and notation of graph theory
- model problems in computer science using graphs and trees
- relate graphs and trees to data structures, algorithms, and counting
- demonstrate and apply advanced counting principles
- discuss and design simple deterministic finite-state machines

Course Synopsis

Advanced counting; Relations; Graphs; Trees; Modelling Computation.

Course Delivery

2 hours lectures, 2 hours lab/tutorial

Modes of Assessment

CA and Final Examination

Reading (and other resources) List

1. Kenneth H. Rosen, "*Discrete Mathematics and Its Applications*", 5th Edition, McGraw-Hill, 2003.
2. Ensley D., and Crawley J., "*Discrete Mathematics: Mathematical Reasoning and Proof with Puzzles, Patterns, and Games* ", John Wiley & Sons, Inc., 2006.
3. Cupillari A., *"The Nuts and Bolts of Proofs"*, Academic Press, 2001.
4. E-resources.

Faculty	Science
Department	Computer Science
Code	CSI141
Title	Programming Principles
Credits	3
Type	Core
Semester in which the course is taught (1 or 2)	1
Pre-requisites	None
Co-Requisites	None

Aims and Objectives

This course aims to introduce students to the basic concepts and principles of programming using a high-level language. Its primary intention is to develop key programming and problem solving skills.

Learning Outcomes

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

- use compilers, editors and debuggers
- develop algorithms to solve problems
- write a program for a given problem
- implement, debug and test programs

Course Synopsis

Overview of problem solving; the programming process; high level languages; data types, input/output, control structures; functions; libraries; file and streams; simple data structures; practical problem-solving exercises.

Course Delivery

2 Lecture hours, 2 hour Lab/Tutorial

Modes of assessment

CA and Examination

Reading (and other resources) list

1. Savitch W., **"***Problem Solving with C++***",** 6th Edition, Addison Wesley, 2006.
2. Horstmann C., "*Big Java*", 3rd Edition, John Wiley & Sons, Inc., 2008.
3. Wu C., *"A Comprehensive Introduction to Object-oriented Programming with Java",* McGraw-Hill, 2008.
4. E-resources.

Faculty	Science
Department	Computer Science
Code	CSI142
Title	Object-Oriented Programming
Credits	4
Type	Core
Semester in which the course is taught (1 or 2)	2
Pre-requisites	CSI141
Co-Requisites	None

Aims and Objectives

This course aims to introduce students to the basic concepts and practice of object-oriented programming using a particular object-oriented programming language and emphasizes on problem solving using the concepts.

Learning Outcomes

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

- Describe the fundamental concepts of object oriented programming and their implementation.
- Design and build object-oriented programs using a modern object-oriented programming language.
- Apply the concepts through a variety of programming projects.

Course Synopsis

Introduction to object-oriented programming (e.g., paradigm, OOP language features); generic programming; classes/objects, encapsulation, messages and methods, inheritance, polymorphism; basic data structures; graphical user interface; exception handling; utilities and packages.

Course Delivery

3 hours Lectures, 3 hour Lab/Tutorial

Modes of assessment

CA and Examination

Reading (and other resources) list.

1. *Wu C., "A Comprehensive Introduction to Object-oriented Programming with Java", McGraw-Hill, 2008.*
2. *Deitel H. and Deitel P., " Java: How to Program", 5th edition, 2003.*
3. *Budd T., " Introduction to Object-Oriented Programming", 3rd Edition, Addison Wesley, 2001 .*
4. E-resources

Faculty	Science
Department	Computer Science
Code	CSI161
Title	Introduction to Computing
Credits	2
Type	Core
Semester in which the course is taught (1 or 2)	1
Pre-requisites	None
Co-Requisites	

Aims and Objectives

This course aims to introduce students to the basic concepts and components of computers, architectures, data representations, software, networks, and the Internet.

Learning Outcomes

After completing the course, students should be able to:

- Discuss the importance of the field of computing
- Describe the functions of the different components of a computer.
- Explain the organization of a computer system
- Describe how data is represented inside a computer
- Perform computer arithmetic
- Describe the different ways of connecting computers for data communications

Course Synopsis

An overview of Computing, logical organization of a computer system, CPU and memory organization, I/O devices characteristics, digital logic: combinational and sequential logic, data representation, computer arithmetic, computer system architecture, computer software, computer network and communication, problem solving using computers, Internet/WWW.

Course Delivery

2 hours Lectures

Modes of assessment

CA and Examination

Reading (and other resources) list:

1. Null L. and Lobur J., "*The Essentials of Computer Organization and Architecture*", 2nd edition, Jones & Bartlett, 2006.
2. Hoganson K., "*Concepts in Computing*", Jones and Bartlet Publishers, 2008.
3. Dale N. and Lewis J., *"Explorations in Computer Science"*, 3rd edition, 2006.
4. E-resources

Faculty	Science
Department	Computer Science
Code	CSI213
Title	Discrete Structures III
Credits	3
Type	Core
Semester in which course is taught (1 or 2)	1
Pre-requisites	CSI132
Co-requisites	none

Aims and Learning Objectives

To introduce students to discrete probability and statistics and their applications in computing. In particular, applications in the areas of algorithm analysis and design, and modelling and simulation.

Learning Outcomes

After completion of the course, students should be able to

- analyze average-case time complexity of basic algorithms
- design randomized algorithms
- apply probabilistic and statistical techniques to model and simulate processes whose input is random

Course Synopsis

Finite probability space; probability measure; events; conditional probability; independence; Bayes' theorem, Integer random variables; expectation; application to algorithm analysis and design; continuous random variables; regression analysis; estimation theory and hypothesis testing; application to modelling and simulation.

Course Delivery

2 hours lectures, 2 hours lab/tutorial

Modes of Assessment

CA and Final Examination

Reading (and other resources) List

1. Dekking F., Kraaikamp C., Lopuhaa H. and Meester L., "A *Morden Introduction to Probability and Statistics: Understanding Why and How*", Springer-Verlag, 2007.
2. Cormen T., Leiserson C., Rivest R. and Stein C., "*Introduction to Algorithms*", 2nd Edition, MIT Press, 2001.
3. E-resources.

Faculty	Science
Department	Computer Science
Code	CSI223
Title	Systems Programming
Credits	3
Type	Core
Semester in which course is taught (1 or 2)	2
Pre-requisites	CSI242
Co-requisites	none

Aims and Learning Objectives

This course introduces students to programming, scripting and software development tools in Unix-like environments.

Learning Outcomes
After completion of the course, students should be able to

- develop, compile, debug and execute applications in Unix-like environments;
- write system programs using many tools that the system provides: commands and library calls;
- use various system tools to maintain code.

Course Synopsis
History of the Unix operating system; System commands; Regular expressions and commands that use them: sed, awk, grep, etc.; Shell programming; Introduction to C programming; Debugging; Low-level input/output; Files and directories; Signals; processes; Interprocess communication; Using make and versioning systems.

Course Delivery
2 hours lectures, 3 hour Labs

Modes of Assessment

CA and Final Examination

Reading (and other resources) List

1. Robbins K. and Robbins S., "*Unix System Programming: Communication, Concurrency, and Threads"*, Prentice Hall, 2003.
2. Rochkind M., *"Advanced UNIX Programming"*, 2nd Edition, Addison-Wesley, 2004.
3. Kernighan B. and Ritchie D., *"The C Programming Language",* Prentice Hall, 1998 .
4. E-resources.

Faculty	Science
Department	Computer Science
Code	CSI241
Title	Programming Fundamentals
Credits	3
Type	Optional/Service
Semester in which course is taught (1 or 2)	1
Pre-requisites Co-requisites	none

Aims and Learning Objectives

This course introduces the principles of problem solving using a computer. Its emphasis will be on teaching students how to build basic user interfaces to perform simple end-user input/output as well as file input/output and using the API.

Learning Outcomes
After completion of the course, students should be able to

- explain the software development process.
- explain what algorithms are and why they are important in programming/computing.
- design and implement programs using GUIs.
- write programs that perform file input/output.
- explain object oriented concepts covered and apply them in designing programs.
- know what APIs are and their importance, as well as how to make effective use of them.
- design and implement and implement programs using iteration, selection, arrays and strings.
- know what exceptions are and how to handle them in their programs.
- design and implement methods.

Course Synopsis
Topics include software development, data types, selection, iteration, algorithms, arrays, strings, object-oriented concepts, methods, exceptions, use of application programming interfaces (APIs), GUIs (graphical user interfaces).

Course Delivery
2 hours lectures, 3 hour lab

Modes of Assessment
CA and Final Examination

Reading (and other resources) List

- Horstmann C., "*Big Java*", 3rd edition, Wiley, 2008.
- Wu C., "*A Comprehensive Introduction to Object Oriented Programming with Java*", McGraw-Hill, 2008.
- E-resources.

Faculty	Science
Department	Computer Science
Code	CSI242
Title	Data Structures
Credits	3
Type	Core
Semester in which course is taught (1 or 2)	1
Pre-requisites	CSI32, CSI142
Co-requisites	none

Aims and Learning Objectives

This course will introduce students to fundamental data structures and data abstraction, algorithm analysis and design paradigms.

Learning Outcomes

After completion of the course, students should be able to

1. describe common applications for each data structure discussed
2. implement user-defined data structures in a high-level language
3. write programs using each data structure discussed
4. choose appropriate data structures for any given problem
5. perform worst-case runtime analysis on data structures and simple algorithms

Course Synopsis

Analysis of algorithms (worst-case); Lists; Stacks and queues; Trees; Graphs; Hashing; Priority queues; Applications of data structures.

Course Delivery

2 hours lectures, 2 hours lab/tutorial

Modes of Assessment

CA and Final Examination

Reading (and other resources) List

1. Weiss M., "*Data Structures and Problem Solving Using Java*", 3rd Edition, Addison Wesley, 2005.
2. Goodrich M. and Tamassia R., "*Data Structures and Algorithms in Java*", 4th Edition, John Wiley and Sons, Inc., 2006.
3. Cormen T., Leiserson C., Rivest R. and Stein C., "*Introduction to Algorithms*", 2nd Edition, MIT Press, 2001.
4. E-resources.

Faculty	Science
Department	Computer Science
Code	CSI243
Title	Functional Programming
Credits	3
Type	Core
Semester in which the course is taught (1 or 2)	1
Pre-requisites	CSI142
Co-Requisites	None

Aims and Objectives

The aim of this course is to introduce students to the main concepts behind the functional programming paradigm and problem solving using functional languages.

Learning Outcomes

After completion of the course, students should be able to

- explain the principles and techniques of programming with functions;
- explain the main features of a modern functional language;
- write non-trivial functional programs;
- develop applications using a functional programming languages

Course Synopsis

Important concepts of functional programming such as recursive definitions, higher-order functions, type inference, polymorphism, abstract data types, modules etc. Programming exercises will illustrate the utility of list-processing, pattern matching, abstraction of data/control, strong typing, and parameterized modules (functors). Mathematical reasoning involved in the design of functional programs and techniques for proving properties about functions so defined.

Course Delivery

2 hours Lectures, 3 hours Lab

Modes of assessment

CA and Examination

Reading (and other resources) list.

1. Thompson S., "*The Craft of Functional Programming*", 2nd Edition, Addison-Wesley, 1999.
2. Bird R., *"Introduction to Functional Programming using Haskell"*, 2nd Edition, Prentice-Hall, 1998.
3. E-resources.

Faculty	Science
Department	Computer Science
Code	CSI251
Title	Computer Architecture and Organization
Credits	3
Type	Core
Semester in which the course is taught (1 or 2)	2
Pre-requisites (if any)	CSI161, CSI141
Co-Requisites (if any)	None

Aims and Learning Objectives
This course introduces to the internal design of a computer, machine level representation of data, assembly level machine organization, storage systems and RAID architectures.

Learning Outcomes
After completion of the course, students will be able to

- discuss the progression of computer architecture from vacuum tubes to VLSI
- demonstrate an understanding of the basic building blocks and their role in the historical development of computer architecture
- use mathematical expressions to describe the functions of simple combinational and sequential circuits
- design a simple circuit using the fundamental building blocks
- discuss, explain and describe the different techniques for representing data
- identify the main types of memory technology
- describe the advantages and limitations of RAID architectures
- explain the organization of the classical von Neumann machine and its major functional units
- explain how an instruction is executed in a classical von Neumann machine
- summarize how instructions are represented at both the machine level and in the context of a symbolic assembler
- explain different instruction formats
- write simple assembly language programs
- demonstrate how fundamental high-level language constructs are implemented at the machine language level
- explain how subroutine calls are handled at the assembly level
- explain the basic concepts of interrupts and I/O operations

Course Synopsis
Digital logic and digital systems; machine level representation of data including representations of records, arrays and nonnumeric data; assemly language programming; storage systems and their technology; RAID architectures.

Course Delivery
2 hours lectures, 3 hour lab

Modes of Assessment
CA and Final Examination

Reading (and other resources) List
1. Jones W., "*Assembly Language Programming for the IBM PC Family*", 3rd Edition, Scott/Jones Inc., 2001.
2. Stallings W., *"Computer Organization and Architecture"*, 5th Edition, Prentice Hall, 2000.
3. E-resources.

Faculty	Science
Department	Computer Science
Code	CSI261
Title	Computer Architecture
Credits	3
Type	Core
Semester in which the course is taught (1 or 2)	2
Pre-requisites (if any)	CSI161
Co-Requisites (if any)	None

Aims and Learning Objectives

This course introduces students to the internal design of a computer, machine level representation of data, storage systems and RAID architectures.

Learning Outcomes
After completion of the course, students will be able to

- discuss the progression of computer architecture from vacuum tubes to VLSI
- demonstrate an understanding of the basic building blocks and their role in the historical development of computer architecture
- use mathematical expressions to describe the functions of simple combinational and sequential circuits
- design a simple circuit using the fundamental building blocks
- discuss, explain and describe the different techniques for representing data
- convert data from one representation to another
- explain the organization of the classical von Neumann machine and its major functional units
- explain how an instruction is executed in a classical von Neumann machine
- identify the main types of memory technology
- describe the advantages and limitations of RAID architectures

Course Synopsis
Overview and history of computer architecture; logic expressions, minimization, sum of product forms; fundamental building blocks(logic gates, flip-flops, counters, registers); machine level representation of data; basic organization of the von Neumann architecture; control unit; instruction fetch, decode, and execution; storage systems and their technology; RAID architectures.

Course Delivery
3 hours lectures

Modes of Assessment
CA and Final Examination

Reading (and other resources) List

- Stallings W., *"Computer Organization and Architecture"*, 5th Edition, Prentice Hall, 2000.
- Hennessy J. and Patterson D., *"Computer Architecture: A Quantitative Approach"*, Morgan Kaufmann, 2006.
- E-resources.

Faculty	Science
Department	Computer Science
Code	CSI262
Title	Database Concepts
Credits	3
Type	Core
Semester in which the course is taught (1 or 2)	2
Pre-requisites	CSI242
Co-Requisites	None

Aims and Objectives:

The aim of this course is to equip students with knowledge and skills in database management, including database requirements analysis and modelling, database design and implementation tools using the relational database technology.

Learning outcomes

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

- design relational databases
- apply normalization to database design
- use Structured Query Language

Course Synopsis

Fundamental principles and concepts of the database systems. DBMS architecture. Databases and data modelling. Services of DBMS. Overview of database languages. Queries, integrity, constraints, aggregate operations. Inference rules for user views. The relational model. Mapping from a conceptual model to a relational model. Database design methodologies.

Course Delivery

2 hours Lectures, 2 hours Lab

Modes of assessment:
CA and Examination

Reading (and other resources) list

1. Thomas C. and Carolyn B., "*Database Systems: A practical Approach to Design, Implementation and Management*", 2nd Edition, Addison-Wesley, 2005.
2. Korth H., Silberschatz A., and Sudarshan S.,*"Database System Concepts",* 5th Edition, McGraw-Hill, 2006.
3. Elmasri R. and Navathe S., *"Fundamentals of Database Systems",* 5th Edition, Addison-Wesley, 2007.
4. E-resources.

Faculty	Science
Department	Computer Science
Code	CSI292
Title	Information Systems Fundamentals
Credits	3
Type	Core
Semester in which the course is taught	1
Pre-requisites	None
Co-Requisites	None

Aims and Learning Objectives

The aim of this course is to introduce students to the basic concepts of Information Systems.

Learning Outcomes

Upon successful completion of this course, students should be able to

- explain organizational uses of information systems
- discuss the importance of IS in organizations
- identify organizational information needs
- use application software to solve organizational information needs.
- discuss the professional and ethical responsibilities of the IS practitioner

Course Synopsis

Systems concepts; information system components and relationships; Perspectives to Organizational IS; competitive advantage of information systems; IS/IT infrastructure; IS development framework; decision making concepts, characteristics of IS professionals and IS career paths; information security, crime, and ethics, Examples of Information Systems, Case studies and practical projects that may include developing macros, designing and implementing user interfaces and reports; developing a solution using application software.

Course Delivery

2 Lecture hours and 2 hour lab/tutorial

Modes of assessment
CA and Examination

Reading (and other resources) list

1. O'Brien J., *"Introduction to Information Systems"*, McGraw-Hill, 2005.
2. Stair R. and Reynolds G., "Fundamentals of Information Systems", 3rd Edition, Course Technology, 2005.
3. E-resources.

Faculty	Science
Department	Computer Science
Code	CSI315
Title	Web Technology and Applications
Credits	3
Type	Core
Semester in which the course is taught	2
Pre-requisites	CSI 262, CSI374
Co-Requisites	None

Aims and Objectives

This course will introduce the techniques and technologies used to develop applications for the World Wide Web on the Internet and Intranets. It will be oriented towards implementation, and will describe and examine the use of a range of key web technologies, both client-side and server-side. It aims to introduce various tools that can offer an efficient approach to implementing various client server solutions.

Learning Outcomes

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

- build a database driven web-based application
- explain internet and web protocols
- discuss security issues and strategies in an enterprise-wide web-based application
- choose appropriate technologies for a particular project

Course Synopsis

Overview of the Internet; Web Protocols; Web application architectures; Client-Server relationships, Web development environments; Web Development Technologies (client-side):- (X)HTML, CSS, JavaScript; Web Development Technologies (server-side):- PHP, MySQL, XML; Web Application Security, Web Application Development Project.

Course Delivery

2 Lecture hours and 3 hours lab

Modes of assessment
CA and Examination

Reading (and other resources) list

1. Xue et al., "*A Web Warrior to Web Programming*", Thompson Publishing, 2003.
2. Umar A., "*Application (Re)Engineering: Building Web-Based Applications and Dealing with Legacies*", Prentice Hall, 1997
3. Shklar L., and Rosen R., "*Web Application Architecture: Principles, Protocols and Practices Protocols and Practices*", Wiley, 2003.
4. E-resources.

Faculty	Science
Department	Computer Science
Code	CSI316
Title	Web Computing
Credits	3
Type	Core
Semester in which the course is taught	2
Pre-requisites	CSI315
Co-Requisites	None

Aims and Objectives

The objectives of the course are to provide students with an appreciation of the major technical and organizational issues related to e-business and e-commerce. It introduces contemporary ways in which companies use the Internet, World-Wide-Web, and other electronic media for conducting and expanding business.

Learning Outcomes

Upon successful completion of the course, each student will be able to:

- Define and discuss major concepts, tools, techniques, and methods of e-business and e-commerce.
- Analyze the strengths and weaknesses of different e-business models.
- Explain technical, ethical, security and policy issues in e-business.
- Analyze how e-business affects the strategic intent and operations of an organization.
- Plan, design, and develop a significant e-commerce project with an application of best practices that are currently available.

Course Synopsis

Introduction to E-Business; Models, Tools, Skills, Business Concepts; E-Business Infrastructures; Technical Foundations, E-Commerce Concepts: Types, Marketing and Distribution. Current practices: Social, Ethical, Security & Legal Issues; Developing Electronic Commerce Applications.

Course Delivery
2 Lecture hours and 2 hours lab

Modes of assessment
CA and Examination

Reading (and other resources) list

1. Laudon K. and Traver C., *"E-Commerce: Business,Technology, Society"* ,4th Edition, Prentice Hall, 2008.
2. E-resources.

Faculty	Science
Department	Computer Science
Code	CSI322
Title	Algorithms
Credits	3
Type	Core
Semester in which course is taught (1 or 2)	1
Pre-requisites	CSI242
Co-requisites	none

Aims and Learning Objectives

To introduce students to the analysis and design of computer algorithms, and to the notion of NP-completeness.

Learning Outcomes

After completion of the course, students will be able to

- argue the correctness of algorithms
- analyze worst-case (occasionally average-case) time complexity of algorithms
- describe the different algorithm design paradigms and synthesize algorithms using these design paradigms
- describe the notion of NP-completeness and its importance. Recite several NP-complete problems and their proofs
- prove that a problem is NP-complete
- synthesize algorithms that employ sorting and data structures as key components
- select appropriate algorithms given the problems context

Course Synopsis

Analysis of algorithms; Review of elementary data structures and algorithms; balance binary search trees; Augmenting data structures; Algorithm design techniques: brute force, backtracking, branch and bound, heuristic, greedy, dynamic programming and divide and conquer; Graph algorithms; NP-completeness.

Course Delivery
2 hours lectures, 2 hours lab/tutorial

Modes of Assessment
CA and Final Examination

Reading (and other resources) List

1. Cormen T., Leiserson C., Rivest R., and Stein C., "*Introduction to Algorithms*", 2nd Edition, MIT Press, 2001.
2. Dasgupta S., Papadimitriou C., and Vazirani U., "*Algorithms*", McGraw-Hill, 2008.
3. Garey M., and Johnson D., "*Computers and Intractability: A Guide to the Theory of NP-completeness*", W. H. Freeman, 1979.
4. E-resources.

Faculty	Science
Department	Computer Science
Code	CSI332
Title	Programming Languages
Credits	3
Type	Core
Semester in which the course is taught (1 or 2)	2
Pre-requisites	CSI243
Co-Requisites	None

Aims and Objectives

To provide in-depth discussion of programming language concepts by focusing on design issues of the various language constructs, examining the design choices for these constructs in a few common languages and critically comparing the design alternatives.

Learning outcomes

After completion of the course, students will be able to

- Explain the basic concepts underlying programming languages
- Compare and contrast the different paradigms and languages and understand the relative advantages and disadvantages of each.
- Select an appropriate language for a particular project

Course Synopsis

Programming Language Concepts. Principles of programming language design. A survey of programming language concepts and design principles of programming paradigms (procedural, functional, object oriented and logic). Topics include a history of programming languages, data types supported, control structures, run-time management of dynamic structures, binding time, representation of data types; data control, sharing, and type checking

Course Delivery
3 lecture hours

Modes of assessment:
CA and Examination

Reading (and other resources) list

1. Sebesta R., "*Concepts of Programming Languages*", 7th Edition, Addison Wesley, 2005.
2. Louden K., "*Programming Languages – Principles and Practice*", 2nd Edition**,** Thomson, 2003.
3. Friedman and Wand, "*Essentials of Programming Languages*", *3rd Edition,* MIT Press, 2008.
4. E-resources

Faculty	Science
Department	Computer Science
Code	CSI341
Title	Introduction to Software Engineering
Credits	3
Type(Core/Optional/Elective/General Education Course)	Core
Semester in which the course is taught (1 or 2)	2
Pre-requisites (if any)	CSI342
Co-Requisites (if any)	None

Aims and Objectives
This course aims to expose students to software development methods and techniques that aid developers in producing quality software systems.

Learning Outcomes
After completing the course, students are expected to:

- Analyze, design, and build software systems.
- Evaluate the quality of an analysis or design.
- Write code to correctly implement a design.
- Verify and validate a software system
- Be able to work in a team

Course Synopsis

Overview of software engineering; Software engineering challenges; development paradigms; process models; requirements analysis; requirements modeling; requirements specification document; system design; software architecture; coding and programming practice; software testing techniques; testing process; software metrics; maintenance issues.

Course Delivery
3 Lecture hours

Modes of assessment
CA and Examination

Reading (and other resources) list.

1. Sommerville I., *"Software Engineering"*, 7th Edition, Addison-Wesley, 2004.
2. Jalote P., "*An Integrated Approach to Software Engineering*", 3rd Edition, Springer, 2005.
3. E-resources.

Faculty	Science
Department	Computer Science
Code	CSI342
Title	System Analysis and Design
Credits	3
Type(Core/Optional/Elective/General Education Course)	Core
Semester in which the course is taught	1
Pre-requisites (if any)	CSI262
Co-Requisites (if any)	None

Aims and Learning Objectives
This course aims to provide students with knowledge and experience of techniques and process modelling associated with the analysis and design of existing or planned systems.

Learning Outcomes
At the end of this course, students should be able to

- describe the processes of systems analysis and design in structured and object-oriented systems development methodologies and life-cycles
- apply techniques commonly used for carrying out the analysis and specification of the design for a computer system
- Develop a complete system using at least one of the methodologies

Course Synopsis

System Analysis and Design Overview, Systems Description and Modelling Techniques, Theory and Methodologies for System Requirements, Analysis and Logical Design including User Interface Design, Physical Design and Implementation, Modern Systems Development tools, Systems Development Management, Cultural, Social and International issues related to systems development, Current Issues in System Development. Project on system analysis and design.

Modes of Assessment
CA and Examination

Course Delivery
2 lecture hours and 2 hours laboratory

Reading (and other resources) list

1. Hoffer J., George J., and Valacich J*., "Modern System Analysis and Design",* 4th Edition, Prentice Hall, 2005.
2. Marakas G, *"Systems Analysis and Design: An Active Approach"*, 2nd Edition, McGraw-Hill, 2006.
3. Dennis A., Haley B., Wixom B. H. and Roth R. M., *"Systems Analysis and Design"*, 3rd Edition, Wiley, 2006.
4. E-resources.

Faculty	Science
Department	Computer Science
Code	CSI344
Title	Artificial Intelligence
Credits	3
Type(Core/Optional/Elective/General Education Course)	Core
Semester in which the course is taught (1 or 2)	2
Pre-requisites (if any)	CSI242
Co-Requisites (if any)	None

Aims and Objectives

This course aims to introduce students to the main concepts and techniques in Artificial Intelligence and practice translating artificial intelligence concepts into working programs.

Learning Outcomes

After completing the course, students should be able to:

- to explain the fundamental concepts of artificial intelligence
- use AI algorithms, techniques and representation methods that can be applied to a wide variety of problems
- develop intelligent systems for computational problems

Course Synopsis

An overview of artificial intelligence, knowledge acquisition, representation, reasoning and inference, search and constraint satisfaction, machine learning techniques and neural networks and their applications.

Course Delivery

2 Lecture hours, 2 hours Lab/Tutorial

Modes of Assessment

CA and Examination

Reading (and other resources) list

1. Russell, and Norvig, "*Artificial Intelligence: A Modern Approach*", 2nd Edition, Prentice Hall, 2003.
2. Poole D., Mackworth A., and Goebel R., *"Computational Intelligence: A Logical Approach",* Oxford University Press, 1998.
3. E-resources.

Faculty	Science
Department	Computer Science
Code	CSI352
Title	Industrial Attachment
Credits	2
Type	Core
Semester in which the course is taught	3 (Winter)
Pre-requisites	CSI354, CSI374, CSI342
Co-Requisites	None

Aims and Learning Objectives

The course is designed with the main objective of giving students an opportunity to work in an organizational environment and to bridge the gap between the classroom and industry. The objectives are to

- provide opportunities for students to gain work experience in the IT related environment by applying the skills the students have acquired during their course of studies
- use the students' experiences in the industry to improve our curriculum
- provide opportunities for us to interact with the industry so as to gain feedback on our curriculum and the performance of our students
- Provide a platform for employment opportunities for our graduates
- To provide organizations with an opportunity to experience the capabilities of our students and assess them as prospective employees.
- To provide an opportunity to establish long term academic-industry links

Learning Outcomes

At the end of this course, student should be able to

- Have a good experience with real-life working environment which would help adapt to the future employment more efficiently.
- Acquire project management skills and team work as they get given projects or researches to carry out on behalf of the organization.
- Demonstrate the knowledge gained through project proposals
- Describe the experiences gained from industrial attachment
- Experience solving real world computer application problems.

Course Synopsis

The students are expected to carry out tasks as specified by the organizations in line with their current level of knowledge. These tasks will vary depending on the type of work the student is performing in the organization. The main expectation is that students be given computing related tasks to perform so as to be able to achieve the objectives of the course.

Modes of Assessment

Student Report, Log sheets, supervision report

Reading (and other resources) list

1. E-Resources.
2. Relevant Company Documents.

Faculty	Science
Department	Computer Science
Code	CSI354
Title	Operating Systems
Credits	3
Type	Core
Semester in which the course is taught (1 or 2)	1
Pre-requisites (if any)	CIS251 or CSI261,CSI242
Co-Requisites (if any)	None

Aims and Learning Objectives
The course describes the fundamental concepts behind operating systems.

Learning Outcomes

- Explain the objectives and functions of modern operating systems.
- Describe the functions of a contemporary operating system with respect to convenience, efficiency, and the ability to evolve.
- Explain the concept of a logical layer.
- Describe how computing resources are used by application software and managed by system software.
- Demonstrate the potential run-time problems arising from the concurrent operation of many separate tasks and describe appropriate solutions.
- Describe commonly used scheduling algorithms.
- Explain the concept of memory and how it is realized in hardware and software.
- Compare and contrast the security strengths and weaknesses of two or more currently popular operating systems with respect to recovery management.
- Compare and contrast different approaches to file organization, recognizing the strengths and weaknesses of each.

Course Synopsis
Overview of operating systems, Operating system principles, Concurrency, Scheduling and dispatch, Memory management, File systems, Security and protection.

Course Delivery
2 lecture hours and 3 hours lab

Modes of Assessment
CA and Examination
.
Reading (and other resources) list

1. Silberschatz, Galvin, and Gagne, *"Operating System Concepts"*, 6th Edition, Wiley & Sons, 2003.
2. Stallings W., "*Operating Systems: Internals and Design Principles*", 5th Edition, Prentice Hall, 2005.
3. Tanenbaum A., and Woodhall A., "*Operating Systems: Design and Implementation*", 3rd Edition, Prentice Hall, 2006.
4. E-resources.

Faculty	Science
Department	Computer Science
Code	CSI374
Title	Computer Networks
Credits	3
Type	Core
Semester in which course is taught (1 or 2)	1
Pre-requisites	CSI142, CSI251 or CSI261
Co-requisites	none

Aims and Learning Objectives
This course introduces computer communication network concepts and the design and implementation of client/server applications using API and remote procedure calls.

Learning Outcomes
After completion of the course, students will be able to

- explain the hierarchical, layered structure of a typical network structure.
- describe the responsibilities of the first four layers of the OSI reference model.
- Discuss the differences between circuit switching and packet switching along with advantages and disadvantages of each.
- explain how a network can detect and correct transmission errors.
- illustrate how a packet is routed over the internet.
- prepare network cables.
- Design and install a simple network using standard configuration tools like DHCP.
- demonstrate the ability to use effectively a range of common networked applications including email, ssh, FTP, newsgroups and instant messaging.
- explain the different roles and responsibilities of clients and servers for a range of possible applications.
- design and implement client/server applications using API and remote procedure calls.
- Describe emerging networking technologies and assess their current capabilities, limitations and near term potential.

Course Synopsis
Introduction to network architectures and protocols; network models; data link layer issues; network layer issues; transport protocols; application layer protocols; routing; network programming; network security and network management.
Course Delivery
2 hours lectures, 3 hour lab

Modes of Assessment
CA and Final Examination
Reading (and other resources) List

1. Forouzan B, "Data Communications and Networking", 4th Edition, McGraw-Hill, 2007.
2. Halsall F., *"Data Communications, Computer Networks and Open Systems"*, 4th Edition, Addison-Wesley, 1995.
3. E-resources.

Faculty	Science
Department	Computer Science
Code	CSI384
Title	Information Systems Theory & Practice
Credits	3
Type	Core
Semester in which the course is taught (1 or 2)	2
Pre-requisites (if any)	CSI292, CSI342
Co-Requisites (if any)	None

Aims and Objectives

To introduce, discuss, and describe fundamental concepts of IS theory and their importance to practitioners to show how an information system is a strategic and integral component of an organization. The course also covers organizational models and their relevance to IS.

Learning outcomes:
At the end of this course, students should be able to:

- explain systems theory, quality, and organizational modeling and demonstrate their relevance to information systems
- discuss how an information system is developed and managed within an organization
- discuss the relevance of the cognitive process and human interactions in information system design and implementation
- discuss the relationship of IS planning to organizational planning
- demonstrate specific classes of application systems including TPS and DSS through practical
- develop a system that demonstrate IS development practices

Course Synopsis

Systems theory and concepts; information systems and organizational system; decision support; level of systems; information systems strategies; roles of information and information technology; roles of people using, developing, and managing systems; IS planning and change management; human-computer interface; IS development process; evaluation of system performance; societal and ethical issues related to information systems design and use.

Course Delivery
2 hour lecture, 2 hours lab

Modes of assessment:
CA and Examination

Reading List

1. Laudon, K.C. and Laudon J.P., "*Essentials of Management Information Systems*", Academic Internet Pub Inc., 2006.
2. Zwass V., *"Foundations of Information Systems"*, McGraw-Hill, 1998.
3. Laudon, K.C. and Laudon J.P., "*Management Information Systems: Managing the Digital Firm*", 10th Edition, Prentice Hall, 2007.
4. E-resources.

Faculty	Science
Department	Computer Science
Code	CSI392
Title	Human Computer Interaction
Credits	3
Type	Optional
Semester in which the course is taught (1 or 2)	2
Pre-requisites (if any)	CSI342
Co-Requisites (if any)	None

Aims and Objectives

The course aims to introduce to students the theory and practice of the design of Computer Interactions Systems and Information Presentation.

Learning Outcomes

After completing the course, students are expected to:

- explain the basic concepts of human computer interaction.
- explain theories and techniques of human computer interaction, and the different models for interaction.
- Carry out requirements analysis and evaluation of real-world interaction systems.
- design and implement appropriate user interfaces for a variety of tasks

Course Synopsis

Basic concepts of human computer interaction; Human abilities; Modelling in user interface design; Design of graphical user interfaces; User-centred design; Task-centred design; Rapid prototyping of interaction systems; Evaluation without users; Evaluation with users; Design evolution.

Course Delivery

2 hours Lectures, 2 hours Lab/Tutorial

Modes of Assessment

CA and Examination

Reading (and other resources) list

1. Dix A., Finlay J., Abowd G., and Beale R., "*Human Computer Interaction*", 3rd Edition, Prentice Hall, 2003.
2. Sharp H., and Rogers Y., and Preece J., "*Interaction Design: Beyond Human-Computer Interaction*", 2nd Edition, Wiley, 2007.
3. E-resources

Faculty	Science
Department	Computer Science
Code	CSI405
Title	Project
Credits	4
Type	Core
Semester in which course is taught (1 or 2)	2
Pre-requisites (if any)	CSI352, CSI315, CSI341
Co-Requisites (if any)	None

Aim and Objectives

This is a capstone project course with the main aim to enable students apply the various knowledge and skill acquired in other courses. The objective is to carry out a problem-solving application project.

Learning outcomes:

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

- develop a system that demonstrates the integration of the various knowledge areas
- define a clear statement of a given problem-solving project;
- Review relevant literature on a given problem-solving project;
- Identify and employ appropriate methodology in carrying out a problem-solving project
- Write a technical quality project report;
- Make professional project presentation;

Course Synopsis

problem definition, literature review, project proposal, identifying and employing appropriate methodology, analysis, design and implementation of appropriate solution; writing quality project report; technical presentation.

Modes of Assessment

Project proposal, design and implementation, presentation, report.

Reading (and other resources) list

1. Literature relevant to project work.
2. E-resources.

Faculty	Science
Department	Computer Science
Code	CSI411
Title	Theory of Computation
Credits	3
Type	Core
Semester in which course is taught (1 or 2)	1
Pre-requisites	CSI322
Co-requisites	none

Aims and Learning Objectives
To introduce students to automata and complexity theory. It will also discuss techniques for dealing with NP-hardness.

Learning Outcomes
After completion of the course, students will be able to

- classify languages as to which Chomsky hierarchy they belong to
- prove that a language is in a specified class and not in the next lower class
- convert between equivalent notations for a language
- define basic complexity classes
- prove that a problem is NP-complete using a reduction from a known NP-complete problem
- describe approximation algorithms for well-known NP-complete problems
- explain the Church-Turing thesis and its significance
- explain the notion of uncomputability
- prove that a problem is uncomputable

Course Synopsis
Deterministic (DFA) and nondeterministic (NFA) finite automata; Equivalence between DFAs and NFAs; Regular expressions; Push-down automata (PDA); Relationship between PDAs and context-free grammars; Pumping lemmas for regular expressions and context-free grammars; Properties of context-free grammars; Turing machines; Chomsky hierarchy; The Church-Turing thesis; NP-completeness; Cook's theorem; Standard NP-complete problems; Reduction techniques; approximation algorithms.

Course Delivery
2 hours lectures, 2 hours lab/tutorial

Modes of Assessment
CA and Final Examination

Reading (and other resources) List

1. Sipser M., "*Introduction to the Theory of Computation*", 2nd Edition, PWS Publishing Company, 2005.
2. Cormen T., Leiserson C., Rivest R., and Stein C., "*Introduction to Algorithms*", 2nd Edition, MIT Press, 2001.
3. E-resources.

Faculty	Science
Department	Computer Science
Code	CSI412
Title	Programming Language Translation
Credits	3
Type	Core
Semester in which course is taught (1 or 2)	2
Pre-requisites (if any)	CSI411
Co-Requisites (if any)	None

Aims and Objectives

The course aims to introduce to students the theory underlying the modern approach to compiler design and shows how to apply this theory to build compilers and interpreters, as well as compiler generators.

Learning Outcomes

After completing the course, students are expected to:

- explain the fundamental issues in language design (lexical analysis, parsing, symbol tables, declaration and storage management, code generation, and optimization techniques, etc.)
- Be able to construct a compiler
- Be able to write large and complex programs

Synopsis

The principles and design aspects of programming language translation. Compiler organisation. Lexical analysis, Syntax analysis, type checking, code generation, optimisation. Alternative parsing strategies, comparison with respect to time and space trade offs. Grammars and ambiguity. Data representation. Error recovery strategies. Symbol table design. Binding. Compiler writing tools, syntax directed editors. Linkers, loaders, incremental compiling, interpreters. Abstract machine concept.

Course Delivery
2 lecture hours, 2 lab hours

Modes of Assessment:
CA and Examination.

Reading (and other resources) List

1. Louden K., "*Compiler Construction: Principles and Practice*", PWS Publishing, 1997.
2. Appel A., "*Modern Compiler Implementation in Java*", 2nd Edition, Cambridge University Press, 2002.
3. E-resources.

Faculty	Science
Department	Computer Science
Code	CSI414
Title	Decision Support Systems
Credits	3
Type	Core
Semester in which the course is taught	8
Pre-requisites (if any)	CSI342,CSI262
Co-Requisites (if any)	None

Aims and Objectives

This course provides an understanding of how computer based systems can be used to support decisions in managerial contexts. It places emphasis on developing the skills necessary to use decision support technologies to enhance the ability to identify problems and develop technical solutions. The course will introduce Decision Support Systems (DSS) technology and applications and provide a set of practical ideas for cost-effective and productive DSS design.

Learning Outcomes

After completing the course, students should be able to:

- describe a decision making process and its application to business problems generally,
- Apply analytical methods and computer technology in finding solutions to business decision problems.
- Model and Design a DSS

Course Synopsis

Decision Making: concepts, principles and models. Structure of the decision problem; Principles and concepts of operation and management support systems; DSS Framework: Data, Model, and Interface components; Decision Support technologies: DSS applications: DSS Model Representation and Methodology; DSS Development Tools; DSS planning, design, implementation and evaluation; Decision model construction; Intelligence and Decision Support Systems; Group support systems; Practical DSS development project.

Course Delivery
2 lecture hrs and 2 hrs laboratory

Modes of Assessment
CA and Examination
Reading (and other resources) list

1. Turban E. and Aaronson J., "*Decision Support Systems and Intelligent Systems*", 7th Edition, Prentice Hall, 2005.
2. Sauter V., *"Decision Support Systems"*, John Wiley, 1997.
3. Marakas G., *"Decision Support Systems"*, Prentice Hall, 2000.
4. E-resources.

Faculty	Science
Department	Computer Science
Code	CSI424
Title	Network Algorithms
Credits	3
Type	Core
Semester in which course is taught (1 or 2)	2
Pre-requisites	CSI374, CS322
Co-requisites	CSI461

Aims and Learning Objectives

The course will discuss algorithms for designing and analyzing network protocols. The emphasis of the course will be on designing and implementing network algorithms and protocols.

Learning Outcomes

After completion of the course, students will be able to

- describe the major algorithms used in packet routing and scheduling
- describe major network protocols: function, advantages and limitations
- describe bandwidth allocation and flow control mechanisms
- describe algorithms for peer-to-peer networking
- design and implement peer-to-peer networks
- analyze network algorithms and protocols

Course Synopsis

Packet routing and scheduling; bandwidth allocation; peer-to-peer networks and their implementation; fairness and quality of service.

Course Delivery
2 hours lectures, 2 hours lab/tutorial

Modes of Assessment
CA and Final Examination

Reading (and other resources) List

1. Black U, "*IP Routing Protocols : RIP, OSPF, BGP, PNNI & CISCO Routing Protocols*", Prentice Hall, 2000.
2. Verma D., "*Legitimate Applications of Peer-to-Peer Networks*", John Wiley & Sons, Inc., 2004.
3. Journal and Conference papers.
4. E-resources.

Faculty	Science
Department	Computer Science
Code	CSI432
Title	Intelligent Systems
Credits	3
Type	Optional
Semester in which the course is taught (1 or 2)	1
Pre-requisites (if any)	CSI342
Co-Requisites (if any)	None

Aims and Objectives

This course aims to provide students with knowledge of advanced intelligent systems concepts, techniques, methods and applications.

Learning Outcomes

After completion of the course, students should be able to

- describe and differentiate between the different paradigms for the construction of intelligent systems, and associate each paradigm with the type of problem and application
- Explain what genetic algorithms are and contrast their effectiveness with the classic
- problem-solving and search techniques
- Describe and contrast the basic techniques for representing uncertainty.
- Explain how an agent differs from other categories of intelligent systems
- Describe the applications of agent theory, to domains such as software agents, personal assistants, and believable agents.
- Articulate the distinction between techniques for information retrieval, language translation, and speech recognition.

Course Synopsis

Advanced search, advanced knowledge representation and reasoning, agents, Natural language processing

Course Delivery

2 lecture hours and 2 hours laboratory

Modes of Assessment

CA and final examination

Reading list

1. Luger G., *"Artificial Intelligence: Structures and Strategies for Complex Problem Solving"*, Pearson, 2001.
2. Norvig R., "*Artificial Intelligence: A Modern Approach*" , 2nd *Edition,* Prentice Hall, 2003.
3. Negnevitsky M., "*Artificial Intelligence, A Guide to Intelligent Systems*", 2nd Edition, Addison-Wesley, 2004.
4. E-resources.

Faculty	Science
Department	Computer Science
Code	CSI441
Title	Requirements Engineering
Credits	3
Type	Core
Semester in which the course is taught (1 or 2)	VII
Pre-requisites (if any)	CSI341
Co-Requisites (if any)	None

Aims and Objectives

This course aims to introduce to students the concepts, techniques, and tools associated with the analysis, specification, and validation of requirements.

Learning Outcomes
After completing the course, students should be able to:

- explain the role of requirements engineering in the development of software-intensive systems and of the various activities involved in requirements engineering

- elicit, model, analyse, manage, and validate system requirements using state-of-the-art requirements modelling techniques and tools

Course Synopsis

Modelling principles, mathematical models and specification languages, types of models (e.g., information modelling, behavioural, structural, functional, domain, etc.), properties of modelling languages, types of analysis (e.g., correctness, quality, consistency, etc.) requirements process, requirements characteristics, requirements management, requirements elicitation, requirements specification, specification languages, requirements validation.

Course Delivery

2 hours Lectures, 2 Lab hours

Modes of Assessment

CA and Examination

Reading (and other resources) list

1. G. Kotonya and I. Sommerville, *"Requirements Engineering: Processes and Techniques"*, Wiley, 2002.
2. I. Sommerville and P. Sawyer, *" Requirements Engineering - A Good Practice Guide"*, Wiley, 2004.
3. Elizabeth Hull, Ken Jackson, Jeremy Dick, *"Requirements Engineering"*, Springer, 2005.
4. E-resources.

Faculty	Science
Department	Computer Science
Code	CSI444
Title	Software Project Management
Credits	3
Type	Core
Semester in which the course is taught (1 or 2)	2
Pre-requisites (if any)	CSI441 or CSI471
Co-Requisites (if any)	None

Aims and Objectives

This course aims to discuss the concepts and techniques that mangers need to create a plan for software development projects using effective estimation methods and to execute that plan with attention to productivity and quality.

Learning Outcomes

After completing the course, students should be able to:

- Develop a comprehensive project plan for a significant development effort
- Measure project progress, productivity and other aspects of the software process
- Effectively estimate costs for development and evolution of a system using several different techniques
- Apply risk management techniques and dynamically adjust project plans
- Use standards for quality, process and project management

Course Synopsis

Overview of project management; effective project management styles; critical factors for project success; unique problems related to software projects; software project management techniques; project planning: decision priority matrix, team organization, staffing, work breakdown, sizing and estimating, scheduling, control methods such as PERT, CPM, risk identification; project execution and control: change control, tracking and reporting, software configuration management.

Course Delivery

2 hours Lectures, 2 Lab hours

Modes of Assessment
CA and Examination

Reading (and other resources) list.

1. Hughes B. and Cotterell M., *"Software Project Management"*, 2nd Edition, McGraw Hill, 2001.
2. Royce W., *"Software Project Management: a Unified Framework"*, Addison-Wesley, 2002.
3. E-resources.

Faculty	Science
Department	Computer Science
Code	CSI446
Title	Information Systems Project Management
Credits	3
Type	Core
Semester in which the course is taught (1 or 2)	2
Pre-requisites (if any)	CSI482
Co-Requisites (if any)	None

Aims and Objectives
Information Systems (IS) project management encompasses the knowledge, techniques, and tools necessary to manage IS development projects. This course discusses the material that IS managers need to create a plan for IS development project, using effective estimation of size and effort, to execute that plan with attention to productivity and quality.

Learning Outcomes
After completing the course, students should be able to:

- demonstrate skills needed to design a project development and implementation plan
- demonstrate skill in use of project management tools and methods within the context of an information systems project
- select the proper project management tools and demonstrate their use
- present and explain the evolving leadership role of information management in organizations
- describe the process for development of information systems policies, procedures, and standards in the organization
- discuss management of time and interpersonal relations

Course Synopsis
Managing the system life cycle: requirements determination, design, implementation; system and database integration issues; network management; project tracking, metrics, and system performance evaluation; managing expectations of managers, clients, team members, and others; determining skill requirements and staffing; cost-effectiveness analysis; reporting and presentation techniques; management of behavioural and technical aspects of the project; change management. Software tools for project tracking and monitoring. Team collaboration techniques and tools.

Course Delivery
2 hours Lectures, 2 Lab hours
Modes of Assessment
CA and Examination
Reading (and other resources) list

1. Olson D., "Introduction to Information Systems Project Management", 2nd Edition, McGraw-Hill, 2003
2. Brandon D., "*Project Management for Modern Information Systems*", IRM Press, 2006.
3. E-resources.

Faculty	Science
Department	Computer Science
Code	CSI451
Title	Sensor Networks
Credits	3
Type	Optional
Semester in which the course is taught (1 or 2)	1
Pre-requisites (if any)	CSI374
Co-Requisites (if any)	None

Aims
The aim of this course is to introduce students to design concepts and technologies associated with Sensor Networks with emphasis on the design, management and operation of sensor networks.

Learning Objectives
On successful completion of the course, the student should be able to:

- explain the different architectures of Sensor networks;
- demonstrate knowledge of wide range of applications of Sensor networks;
- design sensor networks;

Course Synopsis

The course provides a broad coverage of challenges and latest research results related to the design and management of sensor networks. Topics covered include: network architectures; applications and design models- sensor nodes, design challenges, contemporary network architectures, operational and contemporary models, performance metrics and software/hardware setups, Network Bootstrapping- deployment mechanisms, coverage, node discovery protocols, localization schemes, clustering, medium access arbitration, network security and routing protocols, Data dissemination and Routing (query models, in-network data aggregation, robust route setup and coping with energy constraints), Physical and Link layers (Radio energy consumption model, power management, medium access arbitration and optimization mechanisms), Dependability issues - security challenges, threat and attack models, quality of service provisioning, clock synchronization, fault tolerant operation. Project.

Course Delivery
3 hour Lectures, 3 hour Lab/Tutorial

Modes of assessment
CA and Examination
.
Reading (and other resources) list

1. Cauligi S. Raghavendra, Krishna Sivalingam, Taieb M. Znati, *"Wireless Sensor Networks"*, Springer, 2006.
2. Holger Karl, Andreas Wilig, *"Protocols and Architectures for Wireless Sensor Networks"*, Wiley-Interscience, 2007.
3. E-resources.

Faculty	Science
Department	Computer Science
Code	CSI461
Title	Computer Networks & Security
Credits	3
Type	Core
Semester in which the course is taught (1 or 2)	2
Pre-requisites (if any)	CSI374
Co-Requisites (if any)	None

Aim
The aim of this course is to equip students with knowledge and skills necessary to design secure computer networks and administer same from both practical and theoretical perspective.

Learning Objectives
On successful completion of the course, the student will be able to

- demonstrate knowledge of basic computer networks and security concepts, principles and issues;
- understand computer networks and security models;
- design secure computer networks;
- manage a computer networks with adequate security consideration

Course Synopsis

Topics covered include; Overview of computer networks, overview of computer security, system vulnerabilities (administrative, application, and network), cryptography, digital signatures, authentication protocols, network management

Course delivery
2 hours Lectures, 3 hour Lab/Tutorial

Modes of Assessment
CA and Examination

Reading (and other resources) list

1. Panko R., *"Corporate Computer and Network security"*, Prentice Hall, 2003.
2. Pfleeger P. and Pfleeger S., "Security in Computing", 3rd Edition, Prentice Hall PTR, 2002.
3. E-resources.

Code and Title: CSI462 Distributed Computing

Faculty	Science
Department	Computer Science
Code	CSI462
Title	Distributed Computing
Credits	3
Type	core
Semester in which the course is taught (1 or 2)	1
Pre-requisites (if any)	CSI354, CSI374
Co-Requisites (if any)	None

Aims and Objectives

The aim of this course is to equip students with knowledge of distributed computing concepts, paradigm and technologies.

Learning Outcomes

On completion of this course, students should be able to

- explain distributed computing paradigm and associated issues;
- develop software solutions in distributed computing environment;
- analyse and design distributed algorithms;

Course Synopsis

Topics include Characteristics and design issues of Distributed Systems (DS), DS architecture; network, operating system, applications, design and implementation of DS, performance, security and reliability issues.

Course Delivery

2 hours Lectures, 3 hour Labs

Modes of Assessment

CA and Examination

Reading (and other resources) list

1. Coulouris G., Dollimore J., and Kindberg T., *"Distributed Systems, Concepts and Design"*, 4th Edition, Addison Wesley, 2005.
2. E-resources

Faculty	Science
Department	Computer Science
Code	CSI464
Title	Mobile Computing
Credits	3
Type	Core
Semester in which the course is taught (1 or 2)	2
Pre-requisites (if any)	CSI374

Aims and Objectives

The main aim of this course is to cover advanced topics in mobile computing and wireless networks.

Learning Outcomes

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

- explain mobile devices and wireless networks protocols;
- demonstrate practical knowledge of how the wireless networks together with the devices connected to the networks operate;
- differentiate mobile computing from wireless networks;
- discuss challenges and design issues of Nomadic, Cellular and MANET networks;
- develop software solution in a mobile programming environment

Course Synopsis

Introduction to Mobile Computing and wireless networks; wireless technologies; WiMax, WiFi, Bluetooth, ad hoc networks, GSM, 802.11, 802.15, Mobile IP, Mobile Routing. Mobile Device Architectures; Energy Modelling and Management; Mobile Software and Applications.

Course Delivery

2 hours Lectures, 3 hour Lab/Tutorial

Modes of Assessment

CA and Examination

Reading (and other resources) list.

1. Perkins C, Ed., "*Ad Hoc Networking*", Addison-Wesley, Reading, 2000.
2. *E-resources.*

Faculty	Science
Department	Computer Science
Code	CSI471
Title	Software Design
Credits	3
Type(Core/Optional/Elective/General Education Course)	Core
Semester in which the course is taught (1 or 2)	1
Pre-requisites (if any)	CSI341
Co-Requisites (if any)	

Aims and Objectives

This course aims to deal with the issues, techniques, strategies, representations, and patterns used to determine how to design and implement a system.

Learning Outcomes
After completing the course, students should be able to:

- Apply a wide variety of design patterns, frameworks, and architectures in designing a wide variety of software
- Design and implement software using several different middleware technologies
- Use sound quality metrics as objectives for designs, and then measure and assess designs to ensure the objectives have been met

Course Synopsis
Fundamental design issues, design principles, design for quality attributes, design strategies (function-oriented, object-oriented, aspect-oriented, etc), architectural design, human computer interface design, detailed design – design patterns, component design, component and system interface design, design evaluation, design transformation

Course Delivery

2 hours Lectures, 2 Lab hours

Modes of assessment

CA and Examination in ration 1:1

Reading (and other resources) list

1. Gamma E., R. Helm, Vlissides J., "*Design Patterns: Elements of Reusable Object-Oriented Software*", Addison Wesley, 1994.
2. Budgen D., "*Software Design*", 2nd Edition, Addison Wesley, 2003.
3. E-resources.

Faculty	Science
Department	Computer Science
Code	CSI472
Title	Social Informatics
Credits	3
Type	Core
Semester in which the course is taught (1 or 2)	1
Pre-requisites (if any)	CSI352
Co-Requisites (if any)	None

Aims and Objectives

To provide knowledge and skills required in understanding the social, cultural, legal, and ethical context and consequences of IT, anticipate possible issues that can arise from technology in general and IT in particular, and be able to frame alternative responses to possible ethical and social issues

Learning outcomes
At end of this course, students should be able to:

- discuss the cultural, social, legal, and ethical context and consequences of IT;
- discuss possible social and ethical issues that can arise from technology in general and IT in particular;
- Frame alternative responses to possible ethical and social issues
- explain theoretical frameworks for social and ethical analysis of IT in application context;
- Apply social analysis theories and skills to cases;
- discuss issues of IT professionalism;
- practice socially responsible IT professionalism entails

Course Synopsis
Overview of social and ethical issues of IT; Theoretical frameworks for social and ethical analysis of IT; Social and ethical issues of specific IT application systems; Privacy and social control. Safety and reliability. Electronic/Virtual communities; IT Ethics and Professional Responsibilities; Legal aspects of IT: Risks and Liabilities; Intellectual property, Copyright; etc. Current trends and global perspectives in social and professional issues of IT; Social Management of IT; Case studies. practical sessions.

Course Delivery
2 lecture hours, 2 hour lab/tutorial

Modes of Assessment
CA and Examination

Reading List:
1. Rob K., Ed, "*Computerization and Controversy - Value Conflicts and Social Choices*", 2nd Edition, Academic Press, 1996.
2. Schellenberg K., *"Computers In Society"*, University of Michigan--Flint 2005.
3. Duncan L., Ed., *"Internet Ethics"*, McMillan Press Ltd., 2000.
4. E-resources.

Faculty	Science
Department	Computer Science
Code	CSI481
Title	Database Systems
Credits	3
Type	Core
Semester in which the course is taught (1 or 2)	1
Pre-requisites (if any)	CSI262
Co-Requisites (if any)	None

Aim

The aim of this course is to equip students with more advanced principles, tools and techniques for database management with emphasis on distributed relational database system environment.

Learning Objectives

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

- understand more advanced concepts, principles, tools and techniques in database theory and practice;
- design practical database solution for small to medium organizations;
- use practical database technologies in organizational database management problem-solving

Course Synopsis

Database systems development framework; Database planning; Logical Physical database design. Query processing and optimization. Failure and recovery. Concurrency Management. Transaction processing Database security, integrity and control. Distributed database systems. Database Administration; Current topics in database research and best practices;

Course Delivery

2 lecture hours, 2 hours lab

Modes of Assessment

CA and Examination

Reading List

1. Raghu R., "*Database Management System*"; WCB/McGraw-Hill, 1998.
2. Oszu M., "*Principles of Distributed Database Systems*", Prentice Hall, 1999.
3. E-resources.

Faculty	Science
Department	Computer Science
Code	CSI493
Title	Computer Graphics
Credits	3
Type	Optional
Semester in which the course is taught (1 or 2)	1
Pre-requisites (if any)	CSI242
Co-Requisites (if any)	None

Aims and Learning Objectives

This course introduces data structures and algorithms useful for presenting data visually on the computer. The course will cover topics such as graphics theory, graphics technology and graphics programming.

Learning Outcomes
After completion of the course, students will be able to

- distinguish the capabilities of different levels of graphics software and describe appropriateness of each
- Create images and apply transformations on them using standard graphics API
- Implement simple algorithms that perform transformation and clipping on a simple 2D image
- discuss the 3D coordinate system and the changes required to extend 2D transformation operations to handle transformations in 3D
- describe appropriateness of graphics architectures for given applications
- explain the function of various input devices
- compare and contrast techniques for raster graphics and vector graphics
- use current hardware and software fore creating and displaying graphics
- discuss the expanded capabilities of emerging hardware and software for creating and displaying graphics
- explain the operation of Bresenham algorithm for rendering a line on a pixel-based display
- explain the concept and applications of each of the rendering techniques
- demonstrate basic rendering techniques by creating an image using a standard API
- describe how a graphic image has been created

Course Synopsis
Graphics software; Using graphics API; Simple colour models; Homogenous coordinates; Affine transformations; Viewing transformation; Clipping; Graphic systems; Basic rendering.
Course Delivery
2 hours lectures, 2 hour lab

Modes of Assessment
CA and Final Examination
Reading (and other resources) List

- Foley J., Van Dam a., Feiner S., Hughes J., and Phillips R., "I*ntroduction to Computer Graphics*", Addison-Wesley, 1993.
- Hearn D. and Baker M., *"Computer Graphics with OpenGL"*, 3rd Edition, Prentice Hall, 2003.
- E-resources.

Faculty	Science
Department	Computer Science
Code	CSI244
Title	Information Management
Credits Type	3 Core
Semester in which course is taught (1 or 2)	1
Pre-requisites	none
Co-requisites	none

Aims and Learning Objectives

The course explains how IT professional develop, deploy, manage and integrate data and information systems to support the organizations

Learning Outcomes

Upon successful completion of this course, students are expected to:

- Differentiate and use key terms such as: information, data, database, database management system, metadata, and data mining
- Explain the role of data, information, and databases in organizations.
- Explain the advantages of a database approach compared to traditional file processing
- Identify and explain the general types of databases: personal, workgroup, department, enterprise.
- Define data quality, accuracy and timeliness, and explain how their absence will impact
- Describe mechanisms for data collection and their implications
- Explain basic issues of data retention, including the need for retention, physical storage, security.
- Give a brief history of database models and their evolution.

Course Synopsis

Information systems, Properties of data and database systems, Analysis of data, forms and sources, Data collection, Data retention, Data Organization Architecture: Database models, Information backup and recovery, Case studies.

Course Delivery

2 hours Lecture and 2 hours tutorial

Modes of assessment

CA and Examination

Reading (and other resources) list

1. *Michael J. Earl, Information Management: the organisational dimension, Oxford Institute of Information Management, PA Consulting Group*
2. *Introducing information management: the business approach Oxford: Butterworth-Heinemann, 2005. viii, 212 p. ISBN 0-7506-668-4*
3. *Blazewicz, J.; Kubiak, W.; Morzy, T.; Rusinkiewicz, M. (Eds.), Handbook on Data Management in Information Systems, International Handbooks on Information Systems , 2003, IX, 578 p. 157 illus., Hardcover, ISBN: 978-3-540-43893-9*

Faculty	Science
Department	Computer Science
Code	CSI293
Title	Information Technology Fundamentals
Credits	3
Type	Core
Semester in which course is taught (1 or 2)	1
Pre-requisites	none
Co-requisites	none

Aims and Learning Objectives

The course aims is to provide to provide foundation knowledge related to IT covering an overview of the discipline of IT, how it relates to other computing disciplines and its diverse contexts in which it is used and the inherent challenges.

Learning Outcomes

Upon successful completion of this course, students are expected to:

- Outline the history of computing technology
- Explain the relationship between IT and other related disciplines
- Explain how the components of IT system Interrelate
- Describe the role of the IT professionalism
- Illustrate the use of information and communication technologies to solve problems as an IT professional.
- Explain the impact of IT on various application domains

Course Synopsis

History of computing technology, Information Age and Information Society, IT and other disciplines, IT systems model, User Interaction, , Information Assurance and Security, Information and communication Technologies, Adaptability, IT Professionalism, Data versus information, Management of complexity concepts, IT application domains.

Course Delivery

2 Lectures and 2 hour practical

Modes of assessment

CA and Examination in ration 1:2

Reading (and other resources) list

1. Efraim Turban, R. Kelly RainerR. Kelly Rainer, Richard E. Potter, Introduction to Information Technology, Wiley; 3 editions (June 1, 2004), ISBN-10: 0471347809.
2. O'Brien J., *"Introduction to Information Systems"*, McGraw-Hill, 2005
3. Goldschlager, L. and Lister, A., *Computer science: a modern introduction*, Prentice-Hall.
4. E-Resources

Faculty	Science
Department	Computer Science
Code	CSI345
Title	Integrative Programming
Credits	3
Type	Core
Semester in which course is taught (1 or 2)	1
Pre-requisites	CSI242, CSI223, CSI354
Co-requisites	none

Aims and Learning Objectives

This course introduces students to programming techniques and technologies that facilitate integration of systems that support an organization. The course covers scripting languages, various programming languages, application programming interfaces, intersystem communication and architectures.

Learning Outcomes

After completion of the course, students should be able to

- Describe different types of architectures for integrating systems
- Describe how web services are used to integrate disparate applications in an organization
- Describe the role of socket programming in intersystem communication and be able to design, develop and test a TCP/IP socket or Datagram program
- Describe the purpose of message and queuing services
- List commonly used low level data communications protocols (e.g., RS232)
- Use XML and the document object model to integrate and exchange data between systems.
- Design, develop and test a program that parses an XML document.
- Write, debug and test one of: web script, server side script, or operating system script

Course Synopsis
Topics covered include intersystem communication, data mapping and exchange, integrative coding, advanced scripting concepts

Course Delivery
2 hours lectures, 3 hours lab

Modes of Assessment
CA and Final Examination

Reading (and other resources) List

1. Paul J. Perrone, Venkata S.R. "Krishna" R. Chaganti, and Tom Schwenk, *J2EE Developer's Handbook* , Sams Publishing ,ISBN 0-672-32348-6
2. Shahram Khosravi , Professional IIS 7 and ASP.NET Integrated Programming ISBN: 978-0-470-15253-9.
3. e-resources

Faculty	Science
Department	Computer Science
Code	CSI485
Title	Systems Administration
Credits Type	3 Core
Semester in which course is taught (1 or 2)	1
Pre-requisites	CSI354, CSI374
Co-requisites	None

Aims and Learning Objectives

This course equips students with knowledge and skills to design, select, apply, deploy and manage computing systems to support an organization. It covers skills and concepts that are essential to the administration of operating systems, networks, software, file systems, file servers, web systems, database systems, and system documentation, policies, and procedures.

Learning Outcomes

After completion of the course, students should be able to

- Identify, describe and perform systems administration responsibilities on each of the various administrative domains
- Describe administration responsibilities common to the various administrative domains.
- Distinguish between server and client services
- Summarize several methods to push a custom configuration of applications to users.
- Assess a systems ability to continue to meet a given organizational need.
- Explain the need for IT policies and training of users
- Compare and contrast proactive administrative activities and reactive administrative activities.
- Design and deliver training sessions on IT systems and policies to users.

Course Synopsis

Topics covered include administrative activities on different administrative domains and platforms, IT policies, user support and user training on IT policies and systems

Course Delivery
2 hours lectures, 3 hours lab

Modes of Assessment
CA and Final Examination

Reading (and other resources) List

1. AElig; Leen Frisch, *Essential System Administration*, *Tools and Techniques for Linux and Unix Administration*, , Third Edition ,O'Reilly Media
2. Mark Burgess, *Principles of Network and System Administration*, John Wiley and Sons, 2nd, Ed, 2004
3. Thomas A. Limoncelli, Christina J. Hogan, Strata R. Chalup:, *Practice of System and Network Administration,* The (2nd Edition) (9780321492661):
4. E-Resources

Faculty	Science
Department	Computer Science
Code	CSI482
Title	Information Systems Engineering
Credits	3
Type	Core
Semester in which the course is taught (1 or 2)	1
Pre-requisites (if any)	CSI384 or CSI345
Co-Requisites (if any)	

Aims and Learning Objectives:

This course examines the system development and modification process of how to gather information in order to identify problems to be solved, determine system requirements and derive a logical design for a typical information system. Students will be exposed to methods to support each stage of these development processes. The course will also cover issues leading to implementation in emerging systems environments. Students will practice project management and interpersonal skills through a group project.

Learning outcomes

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

- Describe principles of engineering design as applied to information systems
- Carry out a process to determine system requirements and a logical design for an information system.
- Construct Information systems architecture and integration framework
- Discuss Multi-platform Information Systems implementation
- Discuss the relationship between business processes and system integration.
- Explain the key components of a project plan.
- Give examples of current testing standards.
- Describe the major features of an Enterprise Integration Applications (CRM, ERP).

Course Synopsis

Information Systems Engineering concepts and principles; Principles of engineering design for IS; requirements engineering, logical design, physical design and database design; software package evaluation, acquisition, and integration; System implementation; global and inter-organizational issues and system integration; selection of development environments and standards; professional code of ethics; Multi-platform IS; Testing and Quality Assurance, Integration Architecture (CRM, ERP). Project.

Modes of Assessment

CA and Examination

Course Delivery

2hr lecture 2 hr Practical

Reading (and other resources) list

1. Johannesson P., and Soderstrom E., "*Information Systems Engineering: From Data Analysis to Process Networks"*, IGI Publishing, 2008.
2. Langer A., "Analysis and Design of Information Systems", Springer, 2007.

Faculty	Science
Department	Computer Science
Code	CSI452
Title	Information Security Administration
Credits	3
Type	Core
Semester in which the course is taught (1 or 2)	2
Pre-requisites	CSI374
Co-Requisites	None

Aims and Objectives:

The student develops knowledge of the principles of information assurance at the policy, procedural, and technical levels to prepare the student for a role as an IT/IS professional and business decision-maker.

Learning outcomes

After completion of this course, students will be able to:

- Describe the need for and relationship among the attributes of confidentiality, integrity, and availability.
- Describe the McCumber model of information security and use it to describe and evaluate security controls.
- Define the role of policy in driving information security.
- Differentiate among policy, standards, and procedures.
- Describe issue-specific policies and tell how they are used.
- Describe how to identify an organization's information assets.
- Distinguish between identification, authentication, and authorization.
- Describe discretionary and mandatory access control and tell how they are different.
- Enumerate common classes of threats to information assets and describe the technical and procedural protections against each.
- Define annualized loss expectancy (ALE) and describe its role in risk management.
- Describe the concept of layers of information security, and give examples.
- Define and distinguish among incident response plans, disaster recovery plans, business continuity plans, and crisis management plans.
- Discuss business continuity strategies.
- Describe the methods of protecting information in storage and transmission.
- Differentiate between symmetric and public-key cryptography.
- Explain how public-key cryptography can provide for non-repudiation through digital signatures.
- Explain the role of digital certificates in a public-key infrastructure.
- Describe risk assessment for and implementation pf physical security controls.
- Describe the process of maintaining an operating information security plan.

Course Synopsis

Security Fundamentals; Business Needs; Legal, Ethical and Professional Issues; Risk Management; Policies, Standards and Practices; Security Technology: Firewalls and VPNs; Security Technology: Intrusion Detection, Access Control, and Other Security Tools; Cryptography; Physical Security; Implementing Information Security; Information Security Credentials, Security and Personnel; Maintaining Information Security

Course Delivery

2 hours Lectures, 2 hours Lab

Modes of assessment:
CA and Examination

Reading (and other resources) list

1. *Dhillon, Gurpreet, Principles of Information Systems Security: Texts and Cases, WILEY, 2006, ISBN, 9780471450566.*
2. *Gregory B. White, Eric A. Fisch, Udo W. Pooch, Computer System and Network Security, CRC-Press; 1 edition (October 10, 1995) ISBN-10: 0849371791*
3. E-resources.

Faculty	Science
Department	Computer Science
Code	CSI101
Title	Computing Fundamentals
Credits	3
Type	Service
Semester in which the course is taught (1 or 2)	1
Pre-requisites	None
Co-Requisites	

Aims and Objectives

This course aims to introduce students to the basic concepts and components of computers, software, networks, and the Internet.

Learning Outcomes

After completing the course, students should be able to:

- Describe the functions of the different components of a computer
- Explain the organization of a computer system
- Explain the different types of computers
- Describe and apply concepts associated with computer hardware and software
- Describe the different ways of connecting computers for data communications
- Effectively use an operating system

Course Synopsis

An overview of Computing, logical organization of a computer system, CPU and memory organization, I/O devices characteristics, computer software (system & application), computer network and communication, problem solving using computers, Internet/WWW.

Course Delivery

2 Lecture hours and 2 hours lab

Modes of assessment

CA and Examination

Reading (and other resources) list:

1. Hoganson K., "*Concepts in Computing*", Jones and Bartlet Publishers, 2008.
2. Dale N. and Lewis J., *"Explorations in Computer Science"*, 3rd edition, 2006.
3. E-resources

Faculty	Science
Department	Computer Science
Code	CSI102
Title	Programming with C++
Credits Type	3 Service
Semester in which course is taught (1 or 2)	2
Pre-requisites	None
Co-requisites	None

Aims and Learning Objectives

This course introduces the principles of problem solving using a computer. Its emphasis will be on teaching students how to write basic Visual C++ programs. Students will also build basic user interfaces to perform simple end-user input/output as well as file input/output and using the API.

Learning Outcomes
After completion of the course, students should be able to

- design and implement programs using GUIs.
- write programs that perform file input/output.
- explain basic object oriented concepts and apply them in designing programs.
- explain what APIs are and their importance, as well as how to make effective use of them.
- design and implement programs using iteration, selection, arrays and strings.
- write C++ programs that use objects.
- design and use C++ classes.

Course Synopsis
Topics include program development, data types, pointers, selection, iteration, basic algorithms, arrays, strings, functions, classes, use of application programming interfaces (APIs), GUIs (graphical user interfaces).

Course Delivery
2 hours lectures, 2 hour lab

Modes of Assessment
CA and Final Examination

Reading (and other resources) List

- Ivor Horton., "*Ivor Horton's Beginning Visual C++ 2008*", *Big Java*", Wrox, 2008.
- Deitel, P and Deitel H., "*Visual C++.NET: How to Program*", Prentice Hall, 2003.
- E-resources.

Faculty	Science
Department	Computer Science
Code	CSI104
Title	Programming with VB.NET
Credits	3
Type	Service
Semester in which course is taught (1 or 2)	2
Pre-requisites	None
Co-requisites	None

Aims and Learning Objectives

This course introduces the principles of problem solving using a computer. Its emphasis will be on teaching students how to write basic Visual Basic.NET programs. Students will also build basic user interfaces to perform simple end-user input/output as well as file input/output and using the API.

Learning Outcomes
After completion of the course, students should be able to

- explain the software development process.
- discuss the .NET framework.
- design and implement programs using GUIs.
- write programs that perform file input/output.
- explain the basic object oriented concepts and apply them in designing programs.
- explain what APIs are and their importance, as well as how to make effective use of them.
- design and implement programs using iteration, selection, arrays and strings.
- write VB.NET programs that use objects.
- design and use VB.NET classes.

Course Synopsis
Topics include software development, data types, selection, iteration, basic algorithms, arrays, strings, functions, classes, controls and components, use of application programming interfaces (APIs), GUIs (graphical user interfaces).

Course Delivery
2 hours lectures, 2 hour lab

Modes of Assessment
CA and Final Examination

Reading (and other resources) List

- Reynolds M(Editor), et al., "*Beginning VB.NET*", 2nd Edition, Wrox, 2002.
- Deitel, P., "*Visual Basic 2008: How to Program*", Prentice Hall, 2008.
- E-resources.

Faculty	Science
Department	Computer Science
Code	CSI105
Title	Introduction to Web Design
Credits	3
Type	Service
Semester in which course is taught (1 or 2)	1
Pre-requisites	GEC122
Co-requisites	none

Aims and Learning Objectives

This course gives students the theoretical knowledge and practical hands-on skills to develop web sites using XHTML and CSS. The course is aimed at students with no programming background.

Learning Outcomes

After completion of the course, students will have an understanding of the following:

- Internet infrastructure
- Web site planning
- Web browsers and web servers
- XHTML and CSS

Course Synopsis

The Internet and how it works; web design considerations; web site planning; XHTML; CSS; graphics and multimedia.

Course Delivery

2 hours Lecture, 2 hours Lab

Modes of Assessment
CA and Final Examination

Reading (and other resources) List

1. Robbins J, *Learning Web Design: A Beginner's Guide to (X)HTML, StyleSheets, and Web Graphics,* 3rd Ed., O'Reilly Media
2. E-resources.