Sheba University College

Informatics Faculty
Department of Computer Science

Draft Curriculum

For

BSc. in Computer Science

August 2011

Table of Contents

1. Introduction

The Computer Science department of Sheba University College Faculty of Informatics was formed in the 2000 with an initial 4 part-time lecturers and 1 technical assistant. The Department initially introduced diploma in Information Technology in regular and extension program. In the year 2002 the diploma program commenced in the regular program. The diploma program focused on developing practical skills, particularly on the area of database. The last batch of the diploma program graduated in 2004. Since 2002, the department has been offering degree program in Computer Science in both regular and extension admissions. Since its inception, the department grew in both the number of student's intake and number of staff. Currently, it has 21 on-duty staff, 3 on-study leave and six technical assistants. It has currently about 350 students in both regular and extension admissions.

2. Rationale

Computer science is the field of computing that is leading the development of new knowledge in such areas as bioinformatics, intelligent systems, and computer vision and image processing. Computer science is broad field of study spanning theory through to software development. Ethiopia has underscored the use of ICT for development after having taken lessons from other countries. In line with this, the need for research, use and development of software is in demand at both national and regional level. Therefore, the program in computer science will provide the trained manpower that can satisfy the aforementioned professional needs.

3. Objectives of the program

The Computer Science program has both general and specific objectives.

3.1. General objectives

The general objective of Computer Science program is:

- To provide a broad based education in computer science that exposes students to the areas of software engineering, computer architecture, theoretical computer science, systems Programming, computer communications and database management system.
- To enable prospective graduates to acquire the necessary knowledge and skills to meet the growing needs of the country for trained man power in computer science.
- To produce a high quality graduate with an entrepreneurial and problem solving mind set
- To produce highly competent students who are able to practically apply their theoretical knowledge in the real world system.
- To cope up the program with international standards

3.2. Specific objectives

Specifically, the program will enable students to:

- Prove theoretical concepts develop solutions to programming problems
- Design word processor and spreadsheet applications
- Create software interface
- Develop new software systems
- Do system programming
- Design auto-reasoning systems
- Implement intelligent systems
- Design databases
- Implement information retrieval software
- Implement communication software
- Evaluate new forms of search engine
- Produce game and graphics software

4. Professional Profile

Through document analysis on different research organization websites in the field of computer science and based on other universities' computer science programs, it has been found professionals in the field of computer science do have knowledge and skills in software design and implementation, devising new ways of to use computers, developing effective ways to solve computing problems, generating new knowledge through research as in bioinformatics and intelligent systems, presenting findings clearly ,and using computers for the betterment of society.

Areas of Competences	Professional Profile
Essential Skills	▪ Theory and principles of Computing ▪ Software design and implementation ▪ Design of and implementation of auto reasoning systems ▪ Research in computing to generate new knowledge ▪ Ethical use of computer
Skill/tools	▪ Research ability ▪ Using computer applications ▪ Communicating findings clearly
Attitude, Ethics	▪ Develop coherent and ethical personality that includes using computers for good ▪ Promote the use of computers for national and regional developments ▪ Collaborate with other computing fields of study, other research areas such as biotechnology to put knowledge of computer science to currently new knowledge frontiers
Emerging and cross-cutting issues	▪ Addressing controversial issues in robotics, computer ethics
Communication	▪ Reporting orally and/or in writing research findings
Management	▪ Initiate and lead research and projects in computer science

5. Graduate Profile

Graduate profile specifies the expectation of the program at the end of educational process in the department of the computer science. This clearly state what our students able to perform at the end of successful completion of the course work. This mainly focuses on the achievable output unlike the intention described as part of the Professional profile.

5.1. Subject knowledge and understanding

The following are the expected program outputs of the department of computer science in reference to subject knowledge and understanding

1. Knowledge and understanding of the essential facts, concepts, principles and theories related to computer science.
2. Good knowledge of how knowledge and understanding of essential facts, concepts, principles and theories related to computer science can be used to model and design computer systems.
3. Good understanding of how to recognize and critically analyze criteria and specifications appropriate to problems to be solved by computer and plan innovative strategies for their solution.
4. Sound knowledge of the criteria and mechanisms where by computer systems can be critically evaluated and analyzed to determine the extent to which they meet the criteria defined for their current and future development.
5. In depth understanding of the appropriate theory, practices, languages and tools that may be deployed for the specification, design, implementation and evaluation of software systems.
6. Knowledge of how to present succinctly (orally and in written form) rational and reasoned arguments that address a given problem to be solved by computer.
7. Good understanding of the professional, moral and ethical issues involved in the exploitation of computer technology and the associated professional, ethical and legal practices.

5.2. Intellectual skills

This refers to the skills expected at the end of the program from the students of the department. This includes:

1. Analyzing, evaluating and synthesizing skills

- Ability to analyze, synthesizes, evaluate and assess a range of options together with the capacity to apply ideas and knowledge to problems in various situations.

2. Critical reasoning and judgments

- Ability to think logically, reason critically, and make professional judgments.

3. Problem-solving ability

- Ability to apply general principles, concepts, tools techniques and methodologies to solve real world problems
- Effective and efficient problem solving skills including identifying, analyzing and solving problems targeted to user satisfaction.

4. Subject-specific Practical skills

- Analyze, specify, design and develop systems in a manner that is innovative and creative.
- Critically identify, analyze and evaluate risks in a system
- To identify potential securities issues and address them in a system.
- Effectively use recent and up to date tools used for construction and documentation of computer system.
- Ability to troubleshoot and maintain computer system.
- Ability to install, configure and administer computer system.
- Ability to test systems and to address quality assurance issues

5. Research ability

- Ability to understand the basic research methods, and produce technical report and/or research paper in the appropriate scientific style.
- Ability to conduct research, either individually or as part of a team for projects, essays, presentations.
- To lay foundation necessary for advanced study in computer science.

5.3. Transferable skills

1. Communication skills

- Ability to understand and use the English language proficiently, both orally and in writing.
- Effective communication, both oral and in writing, using a range of media in the preparation and presentation of reports.

2. Numeracy and Quantitative skills

- Adequate analytical skills.
- Ability to use, present and evaluate probabilistic and statistical information in a variety of ways.

3. Information and communication technology (ICT) skills

- Customizing ability for different applications
- Basic ability to use ICT related packages
- Ability to use open source packages

4. Interpersonal and Teamwork skill

- Ability to work in groups as a participant who contributes effectively to the group's task.
- Interpersonal skills of effective listening, negotiating, persuasion and presentation
- Ability to critically think and challenge things among the group.

5. Project management skills

- Effective self-management skills in behavior, motivation, and individual initiative.
- Time-management and work-planning skills.
- Team management skills

6. Professional development skills

- The ability to learn effectively for the purpose of continuing professional development and lifelong learning.
- The ability to conduct assigned activities with integrity and professional ethics.
- Ability to develop personal entrepreneur skill.

6. Program Profile

The program is designed to produce graduates which will have skills on integrating computing technologies and their theoretical foundations. The program will have 10 modules. In the modules there are a total of 46 courses. From these 27 are major compulsory courses, 8 are supportive courses, 5 major elective courses, and 6 common courses. The program is designed as a 4 year program based on the time required covering the total 46 courses. The course delivery method encompasses Lecture, Laboratory practice, Assignment (Both written and Practical), Presentations etc.

The mode of evaluation will be vary from course to course but in general it basically includes Projects, Assignments (individual and group), mid and final exam quiz, presentations.

7. Requirements

Here in the following subtopics, we will explain the different requirements.

7.1. Admission Criteria

The admission requirements for the program are similar to the general institutional requirements (Natural + High School Graduates).

7.2. Graduation Requirements

Students must take 37, 35, 35 and 35 credit hours in the 4 years, making a total of 142 credit hours over four years. The Major GPA and CGPA requirements are

Major GPA	2.00	
Overall CGPA	2.00	No F in any course.

For additional requirements the senate legislation should be consulted.

7.3. Nomenclature

English:

"Bachelor of Science Degree in Computer Science"

Amharic:

"¾dÃ"e v‹K` Ç=Ó] u¢Uú¿}` dÃ"e"

7.4. Staff

Currently there are 5 Msc holders (out of which 2 are expatriates), 10 B.Sc. Holders and 5 diploma level Technical assistants.

7.5. Facilities

For this program there are some resources like computer laboratory with local area network, hardware and software resources, and online access facilities. There is also an E-library and department level library.

8. Course Profile

The course is a 4-year programme. In the major courses emphasis is placed on student projects. These projects are intended to give the students a chance to demonstrate their ability to complete a significant computing development task in a chosen area of specialization. Optional courses are included in the third and fourth years which the students may select from a list of courses. To pass the course students must take 37, 35, 35 and 35 credit hours in the 4 years, making a total of 142 credit hours over four years.
The codes for the core courses offered by the computer science department consist of the Four-letter code *COMP*, followed by 3 digits:

- 1st digit: Year (1, 2, 3, or 4. and 5 for elective courses)
- 2nd digit: Subject Area (0 for project work)
- 3rd digit: Semester (Odd number for First and Even number for Second Semesters)

8.1. Major Courses

Course Code	Course Name	Lec Hr	Lab Hr	Credit Hour
COMP121	Introduction to Computer Science	2	2	3
COMP123	Fundamentals of Programming I	2	2	3
COMP124	Fundamentals of Programming II	3	2	4
COMP132	Computer Networking and Data Communication	3	2	4
COMP233	Computer Architecture and Organization	3		3
COMP225	Object-Oriented Programming	3	2	4
COMP261	Fundamentals of Database Systems			4
COMP216	Data Structures and Algorithms	3	2	4
COMP264	Advanced Database Systems	3	2	4
COMP234	Operating Systems	3	2	4
COMP335	Assembly Language Programming	2	2	3
COMP321	Formal Language Theory	3		3
COMP337	Systems Programming	2	2	3
COMP357	Introduction to Artificial Intelligence	2	2	3
COMP342	Complexity Theory	3		3
COMP378	Computer Graphics	3	2	4
COMP328	Advanced Programming	3	2	4
COMP358	Introduction to Neural Networks	2	2	3
COMP439	Internet Programming	3	2	4
COMP401	Professional Ethics in Computer Science	2		2
COMP437	Computer Hardware and Maintenance	3	2	4
COMP417	Analysis of Algorithms	3		3
COMP438	Computer Security	2		2
COMP466	Information Storage and Retrieval	2	2	3
COMP428	Software Engineering	3	2	4
COMP426	Compiler Design	3	2	4
COMP488	Project			3
	Total			**92**

8.2. Supportive Courses

Course code	Course Title	Lec Hrs	Lab Hrs	Credit hours
PHYS131	Digital Electronics	3	-	3
MATH141	Applied Mathematics I	4	-	4
MATH143	Introduction to Probability and Statistics	3	-	3
MATH144	Applied Mathematics II	4	-	4
MATH240	Discrete Mathematics and Combinatorics	3	-	3
MGT202	Entrepreneurship and Small Business Management	3	-	3
MATH248	Numerical Analysis	3	-	3
MATH341	Optimization Theory	3	-	3
	Total			**26**

8.3. Elective Courses

Course Code	Course Name	Lec Hrs	Lab Hrs	Credit Hours
COMP599	Selected Topics in Computer Science			3
COMP597	Expert Systems			3
COMP595	Mobile Computing			3
COMP593	Visual Programming			3
COMP591	Computer Vision and Image Processing			3
	Total			15

8.4. Common Courses

Course code	Course Name	Lec Hr	Lab Hr	Credit Hour
ENG101	Sophomore English	3	-	3
ENG102	Communicative Skills	3	-	3
PHIL102	Introduction to Logic	3	-	3
CIVETH200	Civics and Ethical Education	3	-	3
ECON304	Introduction to Economics	3	-	3
MGT401	Research Methods	3	-	3
	Total			**18**

8.5. Summary of Credit Hours

Subject Area		Credit Hours
Major Area	Compulsory	92
	Elective	6
Supportive	Compulsory	26
Common	Compulsory	18
Total		**142**

8.6. Modules

0. **Social and professional issues**
    a. Sophomore English
    b. Communicative Skills
    c. Introduction to Logic
    d. Civics and Ethical Education
    e. Entrepreneurship and Small Business Management
    f. Introduction to Economics
    g. Professional Ethics in Computer Science
    h. Research Methods
1. **Algorithms**
    a. Data Structures and Algorithms
    b. Analysis of Algorithms
2. **Basics and Programming Languages**
    a. Introduction to Computer Science
    b. Fundamentals of Programming I
    c. Fundamentals of Programming II
    d. Object-Oriented Programming
    e. Formal Language Theory
    f. Advanced Programming
    g. Software Engineering
    h. Compiler Design
3. **Computer System and Networking**
    a. Digital Electronics
    b. Computer Networking and Data Communication
    c. Computer Architecture and Organization
    d. Operating Systems
    e. Assembly Language Programming
    f. Systems Programming
    g. Internet Programming
    h. Computer Hardware and Maintenance
    i. Computer Security
4. **Computational Science**
    a. Applied Mathematics I
    b. Introduction to Probability and Statistics
    c. Applied Mathematics II
    d. Discrete Mathematics and Combinatorics
    e. Numerical Analysis
    f. Optimization Theory
    g. Complexity Theory
5. **Intelligent Systems**
    a. Introduction to Artificial Intelligence
    b. Introduction to Neural Networks
6. **Information Management**
    a. Fundamentals of Database Systems
    b. Advanced Database Systems
    c. Information Storage and Retrieval
7. **Graphics**
    a. Computer Graphics
8. **Project**
    a. Project

9. **Elective Courses**
    a. Selected Topics in Computer Science
    b. Expert Systems
    c. Mobile Computing
    d. Visual Programming
    e. Computer Vision and Image Processing

8.7. Course Breakdown (Regular Program)

Year I Semester I					
Course Code	**Course Name**	**Pre- Requisites**	**Lec Hr**	**Lab Hr**	**Credit Hour**
COMP121	Introduction to Computer Science	-	2	2	3
COMP123	Fundamentals of Programming I	-	2	2	3
PHYS131	Digital Electronics	-	3	-	3
MATH141	Applied Mathematics I	-	4	-	4
MATH143	Introduction to Probability and Statistics	-	3	-	3
ENG101	Sophomore English	-	3	-	3
Total Credit Hours					**19**

Year I Semester II					
Course Code	**Course Name**	**Pre- Requisites**	**Lec Hr**	**Lab Hr**	**Credit Hour**
COMP124	Fundamentals of Programming II	COMP123	3	2	4
COMP132	Computer Networking and Data Communication	-	3	2	4
MATH144	Applied Mathematics II	MATH141	4	-	4
ENG102	Communicative Skills	ENG101	3	-	3
PHIL102	Introduction to Logic	-	3	-	3
Total Credit Hours					**18**

Year II Semester I					
Course Code	**Course Name**	**Pre- Requisites**	**Lec Hr**	**Lab Hr**	**Credit Hour**
COMP233	Computer Architecture and Organization	PHYS111	3	-	3
COMP225	Object-Oriented Programming	COMP124, COMP123	3	2	4
COMP261	Fundamentals of Database Systems	-	3	2	4
MATH240	Discrete Mathematics and Combinatorics	-	3	-	3
CIVETH200	Civics and Ethical Education	-	3	-	3
Total Credit Hours					**17**

Year II Semester II					
Course Code	**Course Name**	**Pre- Requisites**	**Lec Hr**	**Lab Hr**	**Credit Hour**
COMP216	Data Structures and Algorithms	COMP124, COMP123	3	2	4
COMP264	Advanced Database Systems	COMP261, COMP225	3	2	4
COMP234	Operating Systems	COMP233	3	2	4
MGT202	Entrepreneurship and Small Business Management	-	3	-	3
MATH248	Numerical Analysis	-	3	-	3
Total Credit Hours					**18**

Year III Semester I					
Course Code	**Course Name**	**Pre- Requisites**	**Lec Hr**	**Lab Hr**	**Credit Hour**
COMP401	Professional Ethics in Computer Science		2	-	2
COMP378	Computer Graphics	COMP216, COMP225, COMP124, COMP123	3	2	4
COMP439	Internet Programming	COMP132, COMP328, COMP225	3	2	4
COMP437	Computer Hardware and Maintenance	COMP233, COMP234	3	2	4
ECON304	Introduction to Economics	-	3	-	3
Total Credit Hours					**17**

Year III Semester II					
Course Code	**Course Name**	**Pre- Requisites**	**Lec Hr**	**Lab Hr**	**Credit Hour**
COMP335	Assembly Language Programming	COMP233	2	2	3
COMP321	Formal Language Theory		3	-	3
COMP337	Systems Programming	COMP233, COMP234,COMP124, COMP123	2	2	3
COMP357	Introduction to Artificial Intelligence	COMP216, COMP225, COMP124, COMP123	2	2	3
MATH341	Optimization Theory	-	3	-	3
COMP59*	ELECTIVE COURSE	-	-	-	3
Total Credit Hours					**18**

Year IV Semester I					
Course Code	**Course Name**	**Pre- Requisites**	**Lec Hr**	**Lab Hr**	**Credit Hour**
COMP342	Complexity Theory	COMP321	3	-	3
COMP438	Computer Security	COMP337, COMP234, COMP233, COMP132	2	-	2
COMP426	Compiler Design	COMP321, COMP216, COMP225, COMP124, COMP123	3	2	4
COMP466	Information Storage and Retrieval	COMP216	3	-	3
MGT401	Research Methods	-	3	-	3
COMP428	Software Engineering	COMP216, COMP225	3	2	4
Total Credit Hours					**19**

Year IV Semester II					
Course Code	**Course Name**	**Pre- Requisites**	**Lec Hr**	**Lab Hr**	**Credit Hour**
COMP328	Advanced Programming	COMP261, COMP225, COMP124, COMP123	3	2	4
COMP417	Analysis of Algorithms	COMP216	3	-	3
COMP358	Introduction to Neural Networks	COMP357, COMP216, COMP225, COMP124, COMP123	2	2	3
COMP59*	ELECTIVE COURSE	-	-	-	3
COMP488	Project	-	-	-	3
Total Credit Hours					**16**

8.8. Course Breakdown (Extension Program)

Year I Semester I					
Course Code	**Course Name**	**Pre- Requisites**	**Lec Hr**	**Lab Hr**	**Credit Hour**
COMP121	Introduction to Computer Science	-	2	2	3
COMP123	Fundamentals of Programming I	-	2	2	3
PHYS131	Digital Electronics	-	3	-	3
MATH143	Introduction to Probability and Statistics	-	3	-	3
Total Credit Hours					**12**

Year I Semester II					
Course Code	**Course Name**	**Pre- Requisites**	**Lec Hr**	**Lab Hr**	**Credit Hour**
MATH141	Applied Mathematics I	-	4	-	4
ENG101	Sophomore English	-	3	-	3
COMP124	Fundamentals of Programming II	COMP123	3	2	4
Total Credit Hours					**11**

Year I Summer I					
Course Code	**Course Name**	**Pre- Requisites**	**Lec Hr**	**Lab Hr**	**Credit Hour**
COMP132	Computer Networking and Data Communication	-	3	2	4
MATH144	Applied Mathematics II	MATH141	4	-	4
ENG102	Communicative Skills	ENG101	3	-	3
Total Credit Hours					**11**

Year II Semester I					
Course Code	**Course Name**	**Pre- Requisites**	**Lec Hr**	**Lab Hr**	**Credit Hour**
PHIL102	Introduction to Logic	-	3	-	3
COMP233	Computer Architecture and Organization	PHYS111	3	-	3
COMP225	Object-Oriented Programming	COMP124, COMP123	3	2	4
COMP401	Professional Ethics in Computer Science	-	2	-	2
Total Credit Hours					**12**

Year II Semester II					
Course Code	**Course Name**	**Pre- Requisites**	**Lec Hr**	**Lab Hr**	**Credit Hour**
COMP261	Fundamentals of Database Systems	-	3	2	4
MATH240	Discrete Mathematics and Combinatorics	-	3	-	3
COMP216	Data Structures and Algorithms	COMP124, COMP123	3	2	4
Total Credit Hours					**11**

Year II Summer II					
Course Code	**Course Name**	**Pre- Requisites**	**Lec Hr**	**Lab Hr**	**Credit Hour**
CIVETH200	Civics and Ethical Education	-	3	-	3
COMP264	Advanced Database Systems	COMP261, COMP225	3	2	4
COMP234	Operating Systems	COMP233	3	2	4
Total Credit Hours					**11**

Year III Semester I					
Course Code	**Course Name**	**Pre- Requisites**	**Lec Hr**	**Lab Hr**	**Credit Hour**
MGT202	Entrepreneurship and Small Business Management	-	3	-	3
MATH248	Numerical Analysis	-	3	-	3
COMP335	Assembly Language Programming	COMP233	2	2	3
COMP321	Formal Language Theory		3	-	3
Total Credit Hours					**12**

Year III Semester II					
Course Code	**Course Name**	**Pre- Requisites**	**Lec Hr**	**Lab Hr**	**Credit Hour**
COMP337	Systems Programming	COMP233, COMP234,COMP124, COMP123	2	2	3
COMP357	Introduction to Artificial Intelligence	COMP216, COMP225, COMP124, COMP123	2	2	3
MATH341	Optimization Theory	-	3	-	3
COMP59*	ELECTIVE COURSE	-	-	-	3
Total Credit Hours					**12**

Year III Summer III					
Course Code	**Course Name**	**Pre- Requisites**	**Lec Hr**	**Lab Hr**	**Credit Hour**
COMP342	Complexity Theory	COMP321	3	-	3
COMP378	Computer Graphics	COMP216, COMP225, COMP124, COMP123	3	2	4
COMP439	Internet Programming	COMP132, COMP328, COMP225	3	2	4
ECON304	Introduction to Economics	-	3	-	3
Total Credit Hours					**11**

Year IV Semester I					
Course Code	**Course Name**	**Pre- Requisites**	**Lec Hr**	**Lab Hr**	**Credit Hour**
COMP426	Compiler Design	COMP321, COMP216, COMP225, COMP124, COMP123	3	2	4
COMP437	Computer Hardware and Maintenance	COMP233, COMP234	3	2	4
COMP466	Information Storage and Retrieval	COMP216	3	-	3
COMP438	Computer Security	COMP337, COMP234, COMP233, COMP132	2	-	2
Total Credit Hours					**12**

Year IV Semester II					
Course Code	**Course Name**	**Pre- Requisites**	**Lec Hr**	**Lab Hr**	**Credit Hour**
COMP417	Analysis of Algorithms	COMP216	3	-	3
MGT401	Research Methods	-	3	-	3
COMP428	Software Engineering	COMP216, COMP225	3	2	4
COMP328	Advanced Programming	COMP261, COMP225, COMP124, COMP123	3	2	4
Total Credit Hours					**12**

Year IV Summer IV					
Course Code	**Course Name**	**Pre- Requisites**	**Lec Hr**	**Lab Hr**	**Credit Hour**
COMP358	Introduction to Neural Networks	COMP357, COMP216, COMP225, COMP124, COMP123	2	2	3
COMP59*	ELECTIVE COURSE	-	-	-	3
COMP488	Project	-	-	-	3
Total Credit Hours					**12**

9. Description of Courses

9.1. Description of courses- Major

Course Title	**Introduction to Computer Science**
Course Code	Comp121
Credit/Contact Hours	3 Credit Hr (2 Lecture Hrs, 2 Lab Hours)
Year and Semester	1st year 1st Semester
Course Category	Major
Pre-requisite(s)	None
Rationale	This course gives a highlight on the field of Computer Science, specifically what computer systems are and how computer works. This lays a ground for what the field of study encompasses.
Course Description	An overview of Computer Science, the development of computers, data representation, logical organization of a computer system, computer software, computer arithmetic, computer system architecture, computer network and communication, problem solving using computers (Pseudo code, Algorithm representation, Flowchart)
Course Objectives	Upon the completion of the course the students will be able to: ▪ Explain computing in general ▪ Identify computer software and hardware ▪ Perform computer arithmetic ▪ Write Psedocode, draw flowchart ▪ Acquire basic knowledge of computer network and data communication ▪ Use common MS-DOS commands ▪ Have a basic understanding of Windows GUI environment

Course Content

- Basics
  - Why learn about computers?
  - What are computers used for?
  - What is a computer?
  - Information
  - Characteristics of computers

- Applications of computers
- Types of computer
- The Development of Computers
  - Brief history
  - Trends in computer development
- Hardware and Software
  - What are hardware and software?
  - Input/output peripherals
  - Storage devices
  - The system unit
- Representing Data in a Computer
  - Numbering systems
  - Alphanumeric character sets
  -
- Computer Programming
  - Program execution modes
  - Types of programming languages
- Computer viruses

- Introduction to DOS
  - What is an operating system?
  - Types of operating system
  - What is DOS?
  - Functions of DOS
  - Typing a command
  - Directories and files
  - File names
- DOS Commands
  - Getting help
  - Basic commands
  - Commands for file management
  - Commands for directory management
  - Commands for navigating and viewing the file system
  - Disk management commands
  - The active directory and parent directory symbols
  - Wildcards
  - Batch files
  - File attributes
  - System files
- Windows XP
  - The Windows XP Environment
  - Graphical User Interface (GUI)
  - The desktop
  - Windows – basic parts, minimising, maximising & resizing
  - The 'My Computer' icon
    - Formatting floppy disks
    - Managing your files
    - Make new folders
    - Cut/copy & paste files/folders
    - Delete and Renaming files/folders
  - The Recycle Bin
    - Restoring files

        - Emptying
    - Accessories
        - WordPad
        - Paint
        - Calculator
    - Customising your desktop
        - Moving/resizing the taskbar
        - Changing the date/time
        - Changing the background & screen saver
    - Finding files/folders
    - Windows Explorer
    - Anti-virus software
    - Running a virus scan
- Computer Networks
    - Protocols
    - Types of Networks
    - Network Devices
- Computer Programs
    - The Software development life cycle
    - Designing algorithms
        - Pseudo code
        - Flowchart

Delivery Method Lecture, Laboratory, Group work

Course Requirements

- Mid Exam – 25%
- Lab Exercise – 15%
- Assignment – 20%
- Final Exam – 40%

References

- Norton, peter ,"Introduction to Computers" , 6th Edition,2006
- Noel Kalicharan, "An introduction to computer studies", 1988
- Sanjay Sexena, "A first course in computers", 2003
- Stephen Haag, "Computing concepts", 2nd Edition
- Dida Midekso, "Introduction to computer science", 1994
- "Introduction to Computer Science", R. A. Mata-Toledo & P. K. Cushman, Schaum's Outlines, 2000

Course Title	**Fundamentals of Programming I**
Course Code	COMP123
Credit/Contact Hours	3 Credit Hr(2 Lec Hrs and 2 Lab Hrs)
Year and Semester	1st year 1st Semester
Course Category	Major
Pre-requisite(s)	None
Rationale	Problem solving using computer involves a step by step process. This course exposes students to software development process with procedural paradigm and C programming language.
Course Description	Problem Solving using computers, Algorithms Development (pseudo code and flow chart), Program structure, Constants, types, variables etc., Basic data types, Statements, Flow control(conditional and loops), input/output functions, input/output formatting functions.
Course Objectives	Up on the completion of this course students will be able to: ▪ Apply procedural method of solving problems ▪ Know the basic constructs of flowcharts and pseudo codes ▪ Writing, compile, debug and execute c programs ▪ Build interactive C programs ▪ Understand the software development life cycle ▪ Code, compile and debug C programs

Course Content

- Introduction to C Programming
  - Computer Fundamentals
    - Computer organization and hardware
    - Computer software
  - Programming Languages
    - Machine languages
    - Assembly Languages
    - High-Level Languages
    - Evolution of C
  - The C Programming Environment
  - How to use the computer to Run C Programs
    - Accessing the computer
    - Using the editor to prepare Program and Data Files
    - Compiling, Linking, and Executing C Programs
    - Correcting Compilations and Execution errors
  - Your First Program: “Hello World” Program

- Problem-Solving Techniques
    - Introduction:
    - Problem
    - Problem Solving
    - Using computers in Problem solving:
        - The software Development Method
        - Requirements Specification
        - Analysis
        - Design and Representation of Algorithms
            - Pseudocoding
                - Pseudocoding convention
                - Top-Down Stepwise Refinement of algorithms-using pseudocode
            - Flowcharting
                - Flowcharting convention
                - Top-Down Stepwise Refinement of algorithms-using Flowchart
            - Pseudocoding Vs Flowcharting: Which one to Use?
        - Implementation
            - Programming Errors:
                - Syntax, logic/Design, and execution errors
        - Testing and Verification
            - Traversing execution line
        - Program Documentation
            - Documentation importance
- Fundamentals of the C Programming Language
    - Language Character Set and Tokens
        - Reserved Words
        - Indentifiers
        - Constants
        - String literals
        - Punctuators
        - Operators
            - Precedence of Operators
            - Associativity
            - Precedence/Associativity Table
    - The structure of a C Program
        - Program Comments
        - Preprocessor Directives
        - Data Types and Type Declarations
        - Named Constants
        - Statements
        - Compound Statements
    - First Look at Functions: the main() function
    - Building the minimum language subset
        - Introduction to arithmetic assignment and expression statements
        - Standard output : printf
        - Standard input: scanf
    - Preparing C source program Files
        - Typing the source program

        - Style considerations in Typing source programs
    - Program debugging
        - Debugging Syntax errors
        - Debugging Run-time errors
        - Debugging Design errors
- Simple Selections and Repetitions
    - Programming for simple selections
        - Selection statements
        - Two-way selection using the if statement
        - Using relational operators in conditional expressions
        - Using multiple operators in conditional expressions: Precedence of Arithmetic and Relational operators
        - Using Arithmetic expressions as selection criteria in if statements
        - Nested if statements and multi-way selections
    - Programming for controlled repetitions
        - Repetition statements
        - Controlled repetitions using the while statement
    - Style considerations for if and while statements
    - Structured Programming
- Arithmetic Calculations
    - Basic Arithmetic Operators
    - Increment and decrement operators
    - Compound assignment operators
    - Type of arithmetic expressions
        - Explicit type conversions: the cast operator and casting
    - Mathematical library functions
    - Style considerations for arithmetic and assignment expression statements
    - Arithmetic errors and inaccuracies
        - Division by zero
        - Overflow
        - Underflow
        - Integer division
        - Representational inaccuracies
        - Cancellation inaccuracies
    - Automatic Conversions of Numeric Data Types
        - Arithmetic conversions
        - Assignment conversions
    - C Features For Random Number Generation
        - The standard function rand for pseudorandom number generation
        - The standard function srand for random number generation
- Formatted Input and Output
    - Interactive Vs Batch Programs
    - Standard Input-Output Functions
    - Output Formatting
        - Common output formatting requirements for fundamental data types and strings
        - Output formatting for Numeric data
        - Output formatting for Floating-point data
        - Use of output format functions in output formatting
        - Formatting input data
        - Inputting multiple values using scanf
- Complex Selections and Repetitions

- Complex predicates
  - Logical expressions and logical operators
  - The C logical operators
  - Simplifying complex predicates
  - Precedence of logical operators
  - Multiway selection using the if statement with complex predicates
  - Multiway selection using the switch and break statements
  - Bitwise operators
    - ORing, ANDing, and XORing operators
  - Style considerations for Multiway selection structures
- The pretest Repetition structure
  - The for statement for pretest repetition
  - Equivalence of for and while statements
  - Using the for statement for counter-controlled loops
  - Using the for statement for sentinel-controlled loops
  - Checking for incorrect data in a loop and the continue statement
- The posttest repetition structure
    - The do-while statement
    - Use of do-while for counter- and sentinel-controlled loops
    - Use of the break and continue statements in do-while loops
- Nested loops
- Style considerations for repetition statements

Delivery Method Lecture, Laboratory, Group Project

Course Requirements

- Lab Exercise (20%)
- Group Project (30%)
- Final Exam (50%)

References

- "Programming Using the C Language", Hutchinson & Just.
- "C – How To Program", H. M. Deitel & P. J. Deitel, Prentice Hall, 2001.
- "C – The Complete Reference", H. Schildt, McGraw-Hill, 2000

Course Title	**Fundamentals of Programming II**
Course Code	COMP124
Credit/Contact Hours	4 Credit Hr(3 Lec Hrs and 2 Lab Hrs)
Year and Semester	1st year 2nd Semester
Course Category	Major
Pre-requisite(s)	▪ COMP123 - Fundamentals of Programming I
Rationale	A continuation to Fundamentals of Programming I, this course allows students further understand procedural paradigm and C programming language advanced constructs. This better equips student for advanced course, Systems Programming.
Course Description	C functions, parameter passing , arrays, structures, enumerations, top wise step down refinements approach for problems solving in procedural programming, pointer, pointer arithmetic.
Course Objectives	Upon the completion of the course the students will be able to: ▪ Write modular C programming ▪ Know how to design and implement menu-driven programs ▪ Understand how the input/output mechanism works in c++ ▪ Know how to create user defined data types using enumeration and structure constructs in c. ▪ Understand arrays and their common operations of searching, sorting, inserting and deleting elements.

Course Content

- Modular Programming and Functions
  - Elements of Modular Programs
    - Function Definitions
    - Function Calls
    - Function Declarations/Prototypes
  - Structure of Modular Programs
    - Program Structure and Structure Charts
  - Functions that return values under their names
    - The return statement
  - Functions with parameters
    - Defining functions with parameters
    - Calling functions with parameters
    - Correspondence of actual and formal parameters
  - Parameter passing by value

        - Passing strings to function through parameters
    - Parameter passing by pointer
    - Storage classes, scope, visibility, and Duration of variables
        - Storage classes of variables
        - Scope of variables
        - Multiple declarations of variables and visibility
        - Duration of variables
    - Using driver functions to test Modules
        - Menu-Driven Programs
    - Style considerations in Modular Program Design and implementation
- More On Modular Programming and Functions
    - Pointer Variables
        - Declaring Pointer variables
        - Initializing pointer variables
    - Parameter passing by pointers
    - Modular programming with modular with programmer defined libraries
- Arrays
    - Lists and arrays
        - Declaring arrays
        - Initialization of arrays
        - Operations on arrays
        - Arrays, pointers, and pointer arithmetic
        - Passing arrays to functions
        - Using typedef to facilitate program modifiability
    - Dynamic variables and dynamic arrays
        - Dynamic variables
        - Dynamic arrays
    - Common array operations
        - Sorting
            - Bubble sort
            - Selection sort
        - Searching
            - Sequential search
            - Binary search
    - Pointer Arithmetic with arrays
        - sizeof Operator
    - Strings
    - Higher-Dimensional Arrays
        - Declaring Two-Dimensional Array
        - Initialization of Two-Dimensional Array
        - Operations on Two-Dimensional Arrays
        - Passing Two-Dimensional arrays to Functions
- Structures and Enumerations
    - Structures
        - Declaring Structures
        - Declaring Structure Variables
        - Declaring Nested Structures
        - Referencing Structure Members
        - Initializing Structure Variables
        - Operation on Structure Variables
    - Structures as Function parameters

- Enumerated Data Types
  - Declaring enumerated data types and variables
  - Operations on Enumerated variables
  - Using enumerated data types
- Arrays of structures and structures of Arrays
  - Arrays of structures
  - Structures of arrays

- Character Data and Strings
  - Character data
    - Internal representation of character data
    - Collating sequences
    - Automatic conversion of character data
    - Input and output of character data
    - Character-Handling library functions
    - Character Escape sequence
  - Strings revisited
    - Strings and pointers
    - Compile-time initialization of strings
    - Output of string variables
    - Input of string variables
    - Passing strings to functions
  - String Processing
    - Copying strings
    - Comparing strings
    - Computing length of strings
    - Concatenation of strings
    - Searching strings for substrings
    - Tokenizing strings
- Recursion
  - Recursive problems and recursive Functions
  - Tracing recursive valued functions
  - Recursion vs Iteration
- File-oriented Input-Output
  - Fundamentals of C data files
    - C files and Streams
    - Declaring files
    - Naming data files
    - Opening files and the open() function
    - File open verification
    - Closing files and the close() function
    - Checking for end of file and eof() function
  - Processing Text Files
    - Writing text to Text Files
    - Reading data from text files
    - Copying files
  - Random Access files
- Miscellaneous Things
  - Unions
  - Enumerated Types
  - The Preprocessor
  - Including Files

    - Pathnames
    - Preprocessor Constants
    - Preprocessor Macros
    - Use Header Files
- C and the Heap
    - What is the Heap?
    - Dynamic Arrays
    - Using Dynamic Arrays
    - calloc/malloc functions
    - realloc function
    - Allocating Arrays of Arrays
    - Dynamic Data Structures
    - Linking the List

Delivery Method Lecture, Laboratory, Group Project

Course Requirements

- Lab Exercise (20%)
- Group Project (30%)
- Final Exam (50%)

References

- "Programming Using the C Language", Hutchinson & Just.
- "C – How To Program", H. M. Deitel & P. J. Deitel, Prentice Hall, 2001.
- "C – The Complete Reference", H. Schildt, McGraw-Hill, 2000

Course Title	**Computer Networking and Data Communication**
Course Code	COMP132
Credit/Contact Hours	4 Credit Hrs(3 Lec Hrs and 2 Lab Hrs)
Year and Semester	1st Year 2nd Semester
Course Category	Major
Pre-requisite(s)	None
Rationale	Networks enable the sharing of information, Hardware and Software among computers. The networking technologies have different way of transmitting, representing, and securing information sharing. This course acquaintances students to various networking software, hardware and data communication how those can systematically be analyzed and implemented as suited for different contexts.
Course Description	Introduction to computer networks; advantages/disadvantages of networking; LANs and WANs; peer-to-peer vs. server-based; network topologies; networking hardware; data communications; network architecture; access methods; Ethernet & Token Ring; OSI reference model; network protocols; TCP/IP protocol suite; network applications.
Course Objectives	Upon the completion of the course the students will be able to: ▪ To understand what is meant by a computer network, and to be able to describe some of the advantages and disadvantages of using them ▪ To know what is meant by a LAN and a WAN ▪ To understand the differences between peer-to-peer and server-based networks, and know when it is appropriate to use each type ▪ To be able to describe the 4 basic type of network topology, and understand the difference between a physical and logical topology ▪ To know the function of some of the common hardware devices used in computer networking ▪ To understand how digital data can be communicated over a network ▪ To understand the common network access methods, and to understand the operation of some common network architectures such as the Ethernet and Token Ring ▪ To be aware of the common network operating systems and network protocols in use in modern computer networks ▪ To understand the OSI reference model ▪ To be aware of some of the common applications of computer

networks
- To have had practical experience in setting up and administering a Windows 2000/03 network

Course Content

- Introduction
  - What is a computer network?
  - Advantages and disadvantages of networks
  - LANs and WANs
- Network configuration
  - Network components
  - Peer-to-peer vs. server-based
- Network topologies
  - Bus, star, ring, mesh
  - Hybrid topologies
  - Logical vs. physical topologies
- Networking hardware
  - Hubs, switches, routers, bridges, NICs, repeaters, transceivers
  - Cabling: UTP/STP, coaxial, fibre-optic
  - Wireless networking
- Data communications
  - Digital vs. analogue signals
  - Simplex, half-duplex and full-duplex transmission
  - Baseband/broadband transmission
  - Multiplexing (FDM & TDM)
  - Digital transmission
    - NRZ
    - Manchester encoding
    - Synchronous vs. asynchronous communication
  - Analogue transmission
    - Amplitude modulation
    - Frequency-shift keying
  - Error-checking
- Network architecture
  - Access methods
  - Packets
  - Ethernet & token ring architectures
- Network operating systems
  - Multitasking
  - Client software - the redirector
  - Server software
  - Security models
- The OSI reference model

- Network protocols
    - What is a protocol?
    - TCP/IP
- Network applications
    - The Internet
    - The World Wide Web
    - Email

Delivery Method Lecture, Laboratory

Course Requirements

- Mid Exam – 30%
- Lab Exercise – 30%
- Final Exam – 40%

References

- "Data and Computer Communications", W. Stallings, Prentice-Hall, 2000.
- "An Introduction to Computer Networking", K. C. Mansfield Jr. & J. L. Antonakos, Prentice-Hall, 2002

Course Title	**Computer Architecture and Organization**
Course Code	COMP233
Credit/Contact Hours	3 Credit Hrs(3 Lec Hrs)
Year and Semester	2nd Year 1st Semester
Course Category	Major
Pre-requisite(s)	▪ PHYS131 - Digital Electronics
Rationale	Architecture of computer systems dictates how the component parts are modeled to achieve the desire goal. This course allows students know how computer systems work at a lower level. This in turn, allows student how better to make software solutions that better utilize the underlying computer system.
Course Description	Microprocessor architecture, von Neumann architecture, number systems, computer data representation; memory addressing, real mode, protected mode, execution unit, bus interface unit; input output organization: I/O interface, modes of data transfer, interrupts, DMA, IOP and serial communication; memory organization: memory hierarchy, main memory, auxiliary memory, cache memory, virtual memory; digital logic circuits: map simplification, combinational circuits, flip flops, sequential circuits, integrated circuits, decoders, multiplexers, registers, shift registers, binary counters; advanced computer architecture concepts: CISC, RISC, parallel processing, symmetric multiprocessing, clustering.
Course Objectives	Upon the completion of the course the students will be able to: ▪ At the end of the course students should be able to: ▪ Describe the genealogy of microprocessors ▪ Describe the Intel microprocessor used in personal computers ▪ Describe the structure of the personal computer system ▪ Describe the von Neumann programming model ▪ Describe the structure of instruction sets and their effect on registers and memory contents ▪ Use assembly language functionalities to do arithmetic and logical operations, manage program flow control and modular programming ▪ Describe the memory organization of the computer system ▪ Describe and design digital logic circuits ▪ Describe advanced computer architecture concepts

Course Content

- Introduction to the Microprocessor and Computer
    - A historical background
    - The microprocessor-based personal computer system
    - The Intel and Motorola microprocessors
    - Review of number systems
    - Review of computer data representation
- The microprocessor and its architecture
    - Internal microprocessor architecture – micro-architecture
    - Von Neumann architecture
    - Real mode memory addressing
    - Introduction to protected mode memory addressing
    - Execution and bus interface units
- Chapter 3 - Input output organization
    - Peripheral devices
    - Input output interface
    - Asynchronous data transfer
    - Modes of transfer
    - Priority interrupt
    - Direct memory access (DMA)
    - Input output processor (IOP)
    - Serial communication
- Chapter 4 - Memory organization
    - Memory hierarchy
    - Main memory
    - Auxiliary memory
    - Associative memory
    - Cache memory
    - Virtual memory
- Digital logic circuits
    - Review of logic gates and Boolean algebra
    - Map simplification
    - Combinational circuits (half adder, full adder)
    - Flip flops (JK, SR, D)
    - Sequential circuits
    - Integrated circuits
    - Decoders
    - Multiplexers
    - Registers
    - Shift registers
    - Binary counters
- Advanced computer architecture concepts
    - CISC and RISC computers
    - Parallel processing
    - Symmetric multiprocessing
    - Clustering

Delivery Method Lecture, student-lead presentation, observation

Course Requirements

- Mid Exam (25%)
- Assignment (25%)
- Final Exam (50%)

References

- "Introduction to Computer Science", R. A. Mata-Toledo & P. K. Cushman, Schaum's Outlines, 2000
- “Modern Microcomputers”, Erickson & Vonk
- “Computer Essentials”, Hutchinson & Sawyer
- “Computer System Architecture”, M. Morris Mano, Prentice-Hall

Course Title	**Object-Oriented Programming**
Course Code	COMP225
Credit/Contact Hours	4 Credit Hrs(3 Lec Hrs and 2 Lab Hrs)
Year and Semester	2nd Year 1st Semester
Course Category	Major
Pre-requisite(s)	▪ COMP124 - Fundamentals of Programming II ▪ COMP123 - Fundamentals of Programming I
Rationale	Object-Oriented paradigm has recently emerged to alleviate a better management of software development projects in terms of maintenance, reuse, and better handle complexity. This new paradigm views a software system as a set of interacting objects. On this course, students will acquire the basic principles and benefits what Object-Orientation has to offer.
Course Description	Object oriented paradigm concepts (object, classes, data abstraction, encapsulation, inheritance, polymorphism, composition, association, attribute & operations
Course Objectives	Upon the completion of the course the students will be able to: ▪ Compile and run a Java application ▪ Understand the role of the Java Virtual Machine in achieving platform independence ▪ Navigate through the API docs ▪ Use the Object Oriented paradigm in Java programs ▪ Understand the division of classes into Java packages ▪ Use Exceptions to handle run time errors ▪ Select the proper I/O class among those provided by the JDK

Course Content

- Introduction
  - What is Java
  - History
  - The Java Virtual Machine
  - Simple Java Programs
- Language components
  - Primitive Data Types
  - Unicode
  - Documentation
  - Arrays

    - if
    - else if
    - for
    - while
    - switch
    - break
    - continue
    - Multi-Dimensional Arrays
    - Strings
    - StringBuffer
    - Operator
    - Methods
    - Argument Passing - Primitives
    - Argument Passing - Arrays
- Object oriented programming
    - Defining New Data Types
    - Constructors
    - this
    - Data Hiding/Encapsulation
    - Public and Private Members
    - Defining New Data Types revisited
    - Sending an Object to a Function
    - Composition
    - toString
    - Inheritance
    - Protected Members
    - Polymorphism
    - The Object Class
    - Abstract classes
    - Interfaces
    - Class Members
    - Inner Classes
    - Anonymous classes
- Java program structure
    - Downloading the JDK and the API Doc
    - Packages
    - Importing packages
    - Java Standard Libraries
    - java.lang
    - java.lang.Number
    - java.lang.String
    - java.lang.Math
    - java.lang.System
    - java.util
    - java.util.Vector
    - java.util.Stack
    - java.util.Date
    - java.util.BitSet
    - java.util.Hashtable
    - java.util.Enumeration (interface
    - java.util.Properties

    - Other packages
- Exceptions
    - Introduction
    - An Example of Exception Handling
    - Inheritance and Exceptions
    - Exception Methods
    - Developing Your Own Exception classes
    - The finally clause
- I/o in java
    - Introduction
    - The File Class
    - Listing Files in a directory
    - I/O Classes
    - InputStreams
    - File Streams
    - Data Streams
    - Line Input
    - Print Streams
    - Buffered Streams
    - Random Access Files
    - Reading and Writing Objects
    - Readers and Writer Classes
    - File Reader/Writer
    - String Reader/Writer
    - Tokenizer Classes
    - PrintWriter
    - Deprecated Methods
- Object-Oriented Analysis and Design

Delivery Method Lecture, Laboratory, Team work

Course Requirements

- Lab Exercise (20%)
- Project work (40%)
- Final Exam (40%)

References

- "Ivor Horton's Beginning Java 2 JDK 5Th Edition", Ivor Horton, Wiley Publishing, Inc., 2005
- "Thinking in Java, 3rd Edition", Bruce Eckel, Pearson Education, Inc., 2003

Course Title	**Fundamentals of Database Systems**
Course Code	COMP261
Credit/Contact Hours	4 Credit Hrs(3 Lec Hrs and 2 Lab Hrs)
Year and Semester	2nd Year 1st Semester
Course Category	Major
Pre-requisite(s)	None
Rationale	Better decision making is based on processing of factual, existing data. Database Management Systems do the task of centrally storing and manipulating data in Databases. This course acquaintances students on the foundational mathematical concepts, associated language for manipulating data, and how design of databases.
Course Description	Database systems concepts: definition of a database and benefits of database systems. Database Systems Architecture: Internal, conceptual and External level architectures. Relational data model: Conceptual data model- entity, attribute, relationship, and integrity constraints rules. Database design: ER-model, functional dependencies, avoidance of redundancy and normalization. Mapping ER-models to relational tables. Structured Query language- Data Definition Language, Data Manipulation Language; Basics of Relational Algebra and operation;
Course Objectives	Upon the completion of the course the students will be able to: ▪ Understand what a Database System is, and be able to identify its characteristics and applications, ▪ Explain the Different models of database, ▪ Design ER models from specifications and interpret them into relational tables, ▪ Write SQL statements for data creation and manipulation purposes, ▪ Know how to optimize databases to the most efficient form, ▪ Distinguish and use relational model and relational algebra, ▪ Identify and fix the possible problems that may occur in securing data,

Course Content

- Introduction
  - Overview
  - Basics of Database
  - File organization verses Database approach
  - Users and actors of Database System

- Database System Architecture
  - Data models, Schemas, and Instances
  - Over view of data models
  - DBMS Architecture and Data Independence
- The ER Model
  - The high-level conceptual model
  - Entities, Attributes, and Keys
  - Relationships, Associations, and Constraints
  - The ER Diagrams
  - Mapping ER-models to relational tables
- Functional Dependency and Normalization
  - Functional Dependency
  - Normal Forms
- The SQL Language
  - Data Definition Language
  - Data Manipulation Language
  - Basic Queries in SQL
  - Views
- The Relational Data Model and the Relational Algebra
  - The Relational Model Concepts
  - The Relational Constraints and Relational Database Schemas
  - The Relational Operations
- Data Protection
  - Data recovery
  - Concurrency
  - Data Security

Delivery Method Lecture, Laboratory

Course Requirements

- Mid Exam (30%)
- Project works (30%)
- Final Exam (40%)

References

- "Database: Models, Languages, Design", J. L. Johnson, Oxford University Press, 1997.
- "Relational Database Technology", S. Alagic, Springer-Verlag, 1986.
- "Database Systems: The Complete Book", H. Garcia-Molina, J. D. Ullman & J. D. Widom, Prentice-Hall, 2002.
- "Special Edition Using Microsoft Access 2000", R. Jennings, McMillan Computer Publishing, 1999.

Course Title	**Data Structures and Algorithms**
Course Code	COMP216
Credit/Contact Hours	4 Credit Hrs(3 Lec Hrs and 2 Lab Hrs)
Year and Semester	2nd Year 2nd Semester
Course Category	Major
Pre-requisite(s)	▪ COMP124 - Fundamentals of Programming II ▪ COMP123 - Fundamentals of Programming I
Rationale	Different design alternatives for problem solving might arise. Deciding which is efficient, in terms of memory and CPU usage, is crucial. More over, many data structures surface for different problem domains. Strategically choosing the suitable data structures and algorithms to fit a problem domain is a crucial skill. This course allows students know different data structures and when they are suitable to use. It also acquaintances student how to analyze algorithms for efficiency.
Course Description	Static and dynamic data structures; abstract data types; searching algorithms; sorting algorithms; relationship between data structures and algorithms; recursion vs. iteration; introduction to algorithm complexity & complexity classes
Course Objectives	Upon the completion of the course the students will be able : ▪ To understand the importance of effective use of data structures. ▪ To understand the concept of complexity in relation to algorithms, and appreciate how good data structure design can reduce algorithm complexity. ▪ To be familiar with a number of common data structures and have had practical experience of developing an Abstract Data Type using the C programming language. ▪ To understand the concept of recursion as a programming technique, and to know when and how to use it

Course Content

- Data structures
  - Static vs. dynamic
  - Static data structures
    - Arrays
    - Lists
    - Stacks
    - Queues
  - Dynamic data structures

        - Linked lists
        - Doubly linked lists
        - Graphs
        - Stacks
        - Queues
        - Trees
            - Binary trees
        - Graphs
    - Abstract Data Types
        - What are they?
        - Why use them?
        - Examples
- Algorithms
    - Searching algorithms
    - Sorting algorithms
        - Quicksort
        - Bubble sort
        - Insertion sort
        - Merge sort
    - Relationship between data structures and algorithms
    - Recursion
        - What is it?
        - When should you use it?
        - Recursion vs. Iteration
    - Complexity
        - What is algorithm complexity?
        - Complexity classes

Delivery Method Lecture, Laboratory

Course Requirements

- Mid Exam (30%),
- Final Exam (40%)
- Lab Practical (30%)

References

- "Introduction to the Theory of Complexity", Bovet and Crescenzi, Prentice-Hall, 1993
- "Algorithms and Data Structures – An Approach in C", C. F. Bowman, Oxford University Press, 1998

Course Title	**Advanced Database Systems**
Course Code	COMP264
Credit/Contact Hours	4 Credit Hrs(3 Lec Hrs and 2 Lab Hrs)
Year and Semester	2nd Year 2nd Semester
Course Category	Major
Pre-requisite(s)	▪ COMP261 – Fundamentals of Database Systems ▪ COMP225 – Object-Oriented Programming
Rationale	Mirroring the popularity of Object-Orientation, new Object and Object-Relational Database Systems are emerging. Understanding the methods employed by Database management System while resolving concurrent access of data, optimizing queries, and security measures are discusses. This course covers advanced topics that allow to get the most out of Database Management Systems.
Course Description	This course introduces students to the field of advanced database management systems, whose goal is to build object-oriented databases. In this course we will look at concepts of object-oriented and object-related databases, database systems implementation techniques.
Course Objectives	Upon the completion of the course the students will be able to : ▪ Explain the concepts for object-oriented databases. ▪ Get familiar with object database standard, languages and design. ▪ Identify the related issues with object-relational databases. ▪ Explain basics of query optimization. ▪ Get familiar with transaction processing concepts. ▪ To know about database recovery techniques. ▪ Identify the database security issues.

Course Content

- Concepts for Object-oriented Databases
  - Overview
  - Object Identity
  - Object Structure
  - Type Constructors
  - Encapsulation of Operations
  - Methods & Persistence
  - Type Hierarchies & Inheritance
  - Complex Objects
  - Polymorphism
  - Multiple & Selective Inheritance

- Object database standard & languages and design
  - Overview of the Object Model of ODMG
  - The Object Definition Language
  - The Object Query Language
  - Object Database Conceptual Design
  - Examples of ODBMS's
- Object relational and extended relational databases
  - Evolution & Current Trends of Database Technology
  - Object-Relational Features of Oracle 8
  - Implementation and related issues for Extended type systems.
- Query processing and optimization
  - Overview
  - Translating SQL queries into Relational Algebra
  - Using Heuristics in Query Optimization.
- Transaction Processing Concepts
  - Introduction
  - Transaction & System concepts
  - Desirable properties of Transactions
  - Schedules & Recoverability
  - Serializability of Schedules
- Database Recovery Techniques
  - Recovery Concepts
  - Recovery based on deferred update & Immediate update
  - Shadow paging
- Database Security & Authorization
  - Introduction
  - Discretionary Access control based on Granting / Revoking of privileges
  - Multilevel Security

Delivery Method Lecture, Laboratory, Observation

Course Requirements

- Assignment (20%),
- Project work (40%),
- Final Exam (40%)

References

- "Database: Models, Languages, Design", J. L. Johnson, Oxford University Press, 1997.
- "Database Systems: The Complete Book", H. Garcia-Molina, J. D. Ullman & J. D. Widom, Prentice-Hall, 2002.
- "Database Systems Using Oracle: A Simplified Guide to SQL and PL/SQL", N. D. Shah, Prentice-Hall, 2002.
- "SQL Server 2000 – Web Application Developers Guide", C. Utley, McGraw Hill, 2001.
- "Recovery Mechanisms in Database Systems", V. Kumar & M. Hsu, Prentice-Hall, 1998.
- "Principles of Distributed Database Systems", M. Oszu & P. Valduriez, Prentice-Hall, 1999.

Course Title	**Operating Systems**
Course Code	COMP234
Credit/Contact Hours	4 Credit Hrs(3 Lec Hrs and 2 Lab Hrs)
Year and Semester	2nd Year 2nd Semester
Course Category	Major
Pre-requisite(s)	▪ COMP233 – Computer Architecture and Organization
Rationale	This course acquaintances student with principles operating systems employ for i/o, process, memory, security, and file management.
Course Description	What is an operating system, history, evolution, philosophy; the process concept: the thread concept, process scheduling: basic concepts, scheduling criteria, scheduling algorithms, inter-process communication, process synchronization, the critical section problem, semaphores, monitors, classical synchronization problems; deadlocks: avoidance, prevention, detection; memory management: physical and virtual memory, swapping, allocation, paging, segmentation; file systems: access methods, directory structure, file system implementation, disk space management; Input/Output: principles of I/O hardware and software; security: authentication, encryption.
Course Objectives	Upon the completion of the course the students will be able : ▪ Understand the meaning and history of operating systems ▪ Describe a process and a thread and the role of processes and threads ▪ Describe common inter-process communication and synchronization methods ▪ Describe common process scheduling algorithms ▪ Understand the problem of deadlocks ▪ Describe the implementation of virtual memory as used in computer systems and some of the critical problems that need to be considered ▪ Describe the main issues of operating systems in handling I/O devices ▪ Explain the goals of file system design and the ways in which several operating systems meet these goals ▪ Discuss the need for security in computer systems in the historical context and discuss several threats and methods of overcoming those threats
Course Content	

- Introduction to operating systems
  - Definition of an operating system
  - History of operating systems – computer generations
  - Operating system services
  - Types of operating systems and operating system structure
- Processes and process management
  - The process concept
    - Sequential and concurrent processes
    - Process creation and termination
    - Process states and process state transitions
    - Implementation of processes - PCB
  - The threads concept
    - Multithreading and thread usage (why do we need threads?)
  - Inter-process communication
    - IPC and the race condition
    - Critical sections (or regions) and mutual exclusion
      - Disabling interrupts
      - Lock variables
      - Strict Alternation
      - Peterson's solution
      - The TSL instruction
      - Sleep & wakeup
      - Semaphores (binary and counting) and mutexes
      - Monitors
  - Process scheduling
    - Introduction
      - Scheduling and device queues
      - Scheduling levels: long-term and short-term
      - Context switching
      - Scheduling criteria
      - Process behavior: CPU-bound and I/O bound
    - - Scheduling Algorithms
      - Preemptive and non-preemptive algorithms
      - First-Come-First-Served (FCFS) Scheduling
      - Shortest-Job-First (SJF) and Shortest-Remaining-Time-First (SRTF) algorithms
      - Round-Robin (RR) scheduling
      - Priority Scheduling
        - Dynamic priorities or aging
      - Multilevel Queue Scheduling
        - Multilevel-Feedback-Queue Scheduling
  - Deadlocks
    - Preemptable and non-preemptable resources
    - Definition and conditions for deadlocks
    - Deadlock modeling (Resource Allocation graph)
    - The Ostrich algorithm
    - Deadlock detection and recovery
    - Recovery from deadlock: preemption, rollback, killing processes
    - Deadlock avoidance: safe and unsafe states
    - Deadlock prevention

- Memory management
  - Basic memory management
    - Monoprogramming without swapping and paging
    - Multiprogramming with fixed partitions
      - Fragmentation (internal and external), memory compaction
    - Binding of instructions and data to memory
    - Relocation and protection
  - Multiprogramming with variable partitions - swapping
    - Data structures to keep track of memory: bit maps and linked lists
    - Memory allocation algorithms: first fit, next fit, best fit, worst fit, quick fit
  - Virtual memory
    - Overlays and virtual memory
    - Paging
      - MMU, how does the MMU work?
      - Page tables and problems of page tables: multilevel page tables
      - Structure of a page table entry
      - TLBs (Translation Lookaside Buffers) or associative memory
      - Inverted page tables
    - Page replacement algorithms (PRA)
      - The optimal PRA
      - First-In-First-Out (FIFO) PRA
      - The Second Chance PRA
      - The Clock PRA
      - Least Recently Used (LRU) PRA
      - Least (not) Frequently Used PRA
  - Shared pages – a design issue
  - Segmentation
    - Implementation of pure segmentation
    - Segmentation with paging
- File systems
  - Introduction to file systems
    - The user's point of view
    - Implementation point of view
  - Files
    - Naming, structure, types, access, attributes, operations
    - Memory mapped files
  - Directories or folders
    - Single-level, two-level, and hierarchical directory systems
    - Path names
    - Directory Operations
  - File system implementation
    - File System Layout
    - Implementing Files
      - Contiguous and linked list allocation
      - Allocating using FAT (File Allocation Table)
      - I-Nodes
    - Implementing directories
    - Shared Files
    - Disk space management
      - Choosing a block size

- Keep tracking of free blocks (bit maps and linked lists)
- Disk scheduling algorithms
    - First-Come-First-Served (FCFS) and Shortest Seek First (SJF)
- File system reliability
    - Bad block management, backups, and file system consistency

- Input/Output management
    - Principles of I/O hardware
        - I/O devices
        - Device controllers
        - Memory-mapped I/O
        - Direct Memory Access (DMA)
    - Principles of I/O Software
        - Goals of I/O Software
        - How to perform I/O: programmed I/O, interrupt-driven I/O, using DMA
        - I/O software layers
- Security and protection
    - Threats, intruders, accidental data loss, encryption, user authentication
    - Other security issues: Trojan horse, viruses, worms, logic bomb

Delivery Method Lecture, Laboratory, Student-lead presentation

Course Requirements

- Mid Exam (30%),
- Assignment (20%)
- Final Exam (50%)

References

- "Operating Systems",
  W. Stalling,
  Prentice-Hall, 2001
- "Red Hat Linux – The Complete Reference",
  R. Petersen,
  McGraw-Hill, 2001.
- "Special Edition Using Red Hat Linux",
  A. Simpson & J. Ray,
  Que Publishing, 2001.

Course Title	**Assembly Language Programming**
Course Code	COMP335
Credit/Contact Hours	3 Credit Hours(2 Lec Hrs and 2 Lab Hrs)
Year and Semester	3rd Year 1st Semester
Course Category	Major
Pre-requisite(s)	▪ COMP233 – Computer Architecture and Organization
Rationale	Low level programming languages like Assembly, still find a niche for high speed, real-time software developments; despite the popularity of high-level programming languages. This course equips students with Assembly language practical skills.
Course Description	Overview of Registers and Memory organization, Assembly program structure, Variable declaration and memory allocation, Assembly instructions(data movement, arithmetic, comparison, flow control), Stack operations, Interrupts, Memory Reference techniques, Modular programming, File processing
Course Objectives	Upon the completion of this course the students will be able to:- ▪ Understand assembly language constructs for working with registry, memory allocation, memory reference techniques, File processing, modular programming etc… ▪ Write small to medium scale programs using Assembly Language.

Course Content

- Basics
  - History
  - Comparison of assembly and high level languages
  - Nature of assembly language
- kinds of processors
  - Complex instruction set computers (CISC)
  - Reduced instruction set computers (RISC)
  - Hybrid processors
  - Special purpose processors
  - Hypothetical processors
- data representation
  - size
  - endian
  - Number representations
    - Integer representations

        - Sign magnitude
        - One's complement
        - Two's complement
        - Unsigned
    - floating point representations
- Address space and register set
    - Accumulators
    - Data registers
    - Address registers
    - General purpose registers
    - Constant registers
    - Floating point registers
    - Index registers
    - Base registers
    - Control registers
    - Program counter (location counter)
    - Processor flags
        - Result flags
        - Control flags
    - Stack pointer
    - Subroutine return pointer
- Address modes
    - Absolute address
    - Immediate data
    - Inherent address
    - Register direct
    - Register indirect
        - Address register indirect
        - Address register indirect with postincrement
        - Address register indirect with predecrement
        - Address register indirect with preincrement
        - Address register indirect with postdecrement
        - Address register indirect with displacement
    - Register indirect with index register
        - Address register indirect with index register
        - Address register indirect with index register and displacement
        - Absolute address with index register
        - Memory indirect
        - Memory indirect post indexed
        - Memory indirect pre indexed
    - Program counter indirect
        - Program counter indirect with displacement
        - Program counter indirect with index and displacement
        - Program counter memory indirect postindexed
        - Program counter memory indirect preindexed
- Executable instructions
    - Data movement
    - Address movement
    - Integer arithmetic
    - Floating arithmetic
    - Binary coded decimal

- Advanced math
- Data conversion
- Logical
- Shift and rotate
- Bit manipulation
- Character and string
- Table operations
- High level language support
- Assembly/high level language interface
- Program control
- Condition codes
- Input/output
  - MIX devices
- System control
- Coprocessor and multiprocessor

- File access

Delivery Method Lecture, Laboratory

Course Requirements

- Lab Practical's (20%)
- Project Work (30%)
- Final Exam (50%)

References

- George W. Struble, "Assembler Language Programming", 2000
- Sara **,"**VAX-11 Assembly Language Programming**",** 1999

Course Title	**Formal Language Theory**
Course Code	COMP321
Credit/Contact Hours	3 Credit Hrs(3 Lec Hrs)
Year and Semester	3rd Year 1st Semester
Course Category	Major
Pre-requisite(s)	None
Rationale	Mathematical foundations underpin the design of compilers. This course enables students acquire this.
Course Description	Overview of Natural vs. Formal Language, Review of set theory and relations, Grammars, Introduction to phrase structure grammar and language, Regular grammars, Definition of regular languages and properties, Finite state automata, Definition of DFSA,NFSA, Equivalence of DFSA and NFSA, Regular expressions, Definition, simplification , equivalence between DFSA,NFSA and RE, Pumping Lemma for regular language, Context free grammars, Definition, derivation tree, CFL, ambiguity of grammars, Parsing arithmetic expression, Normal forms(CNF, pumping lemma for CFLs ,GNF), Pushdown automata(NPDA, DPDA)
Course Objectives	Upon the completion of the course the students will be able to: ▪ Understand the difference between natural and formal language ▪ Understand regular expressions ,grammar, and language ▪ Understand finite state automata ▪ Identify DFSA and NFSA ▪ Understand context free grammar ▪ Know how to parse arithmetic expression

Course Content

- Mathematical Foundations
  - Universal programming concepts Relations, functions, cardinality, recursive definitions, proof techniques
  - Language and Grammars
  - Foundations of language theory, string generation, sets
  - Regular languages and expressions
  - Formal languages and grammars
  - Context free grammars
  - Derivations and derivation trees

- Languages from grammars
- Grammars from languages
- Regular grammars
- Regular expressions from regular grammars
- Regular grammars from regular expressions
- Equality and properties of languages
- Programming language syntax: BNF and Algol 60 specification
- Ambiguity and inherent ambiguity
- Normal forms
- Deterministic parsing and Search
- Breadth-First, Top-Down
- Depth-First, Top-Down
- Breadth-First, Bottom-Up
- Depth-First, Bottom-Up

- Finite Automata
  - Deterministic finite automata
  - Representations Acceptors
  - Properties of DFA languages
  - Nondeterministic finite automata
  - Acceptors
  - Properties of NFA languages
  - Removing nondeterminism
  - Equivalence of regular languages, NFAs, and regular expressions
  - Proving regularity or non-regularity of languages (Pumping Lemma)
  - Minimization of DFAs
- Push down automata
  - Acceptance
  - Context free grammars to PDAs
  - PDAs to context free grammars
  - Pumping Lemma
  - Closure properties

Delivery Method Lecture

Course Requirements

- Mid Exam (30%)
- Assignment (20%)
- Final Exam (50%)

References

- Sudkamp, Thomas A.," *Languages and Machines"*, 3rd Edition, Addison Wesley, 2006.

Course Title	**Systems Programming**
Course Code	COMP337
Credit/Contact Hours	3 Credit Hrs(2 Lec Hrs and 2 Lab Hrs)
Year and Semester	3rd Year 1st Semester
Course Category	Major
Pre-requisite(s)	▪ COMP233 – Computer Architecture and Organization ▪ COMP234 – Operating Systems ▪ COMP124 – Fundamentals of Programming II ▪ COMP123 – Fundamentals of Programming I
Rationale	Distributed and real-time application developments need a humble reliance on the hosting operating System. This course enables students how to System program using C.
Course Description	Basic concepts of operating systems and system programming. Processes and inter process communication/synchronization. Virtual memory, program loading and linking. File and I/O subsystems. Utility programs. Study of a real operating system. Introduction to computer systems from a software perspective. Systems programming using C programming language. Representation of data. Pointers and structures and their machine-level representation. Linking and loading. Program debugging.
Course Objectives	Upon the completion of the course the students will be able to: ▪ Understand the concept of operating systems and system programming ▪ Write system programs using C Programming language

Course Content

- Basics
  - Universal programming concepts
  - Syntax
  - How they relate to shell programming
- Shell functionality in regards to
  - Command interpretation
  - Filename expansion
  - Redirection and Pipes
- Quoting mechanisms
- Entering program code and commands
- Shell program execution procedures
- Documenting a program with comments

- Program debugging mechanisms - The set command
- Understanding sub-shells in relationship to program execution
- Functions - The shell's sub-routines
- Command aliases - The alias command
- Shell parameters and variables
    - Declaring variables
    - Using variable values
    - Special shell variables
    - Conditional parameter substitution
    - Positional parameters
    - Using the let command to do simple math

- Shell I/O
- Understanding file descriptors
- The here document
- Terminal control using the tput command
- Conditional testing
    - The test command
    - The if - then - else construct
    - The case construct
- The eval command
- Using the let command to do simple math
- Return codes and values
- Looping Mechanisms
    - The for loop
    - The while loop
    - The until loop
- The select construct
- true and false statements
- break and continue statements
- Signals and Traps
    - Understanding signals
    - Trapping signals
    - Releasing traps
- Programming Considerations
    - Shell programming development cycle
    - Optimizing execution time
    - Programming best practices

Delivery Method Lecture and Laboratory

Course Requirements
- Assignment (30%),
- Project Work (30%)
- Final Exam (40%)

References
- Cameron Newham, Bill Rosenblatt, "Learning the Bash Shell",O'Reilly, 2005

Course Title	**Introduction to Artificial Intelligence**
Course Code	COMP357
Credit/Contact Hours	3 Credit Hrs(2 Lec Hrs and 2 Lab Hrs)
Year and Semester	3rd Year 1st Semester
Course Category	Major
Pre-requisite(s)	▪ COMP216 – Data Structures and Algorithms ▪ COMP225 – Object-Oriented Programming ▪ COMP124 – Fundamentals of Programming II ▪ COMP123 – Fundamentals of Programming I
Rationale	This course is about what underlines an intelligent system design.
Course Description	Introduction to artificial intelligence (AI); history of AI; the Turing test; game playing; expert systems; uncertainty & probabilistic reasoning; machine learning; overview of neural networks; philosophical implications of AI: nature of intelligence and consciousness, can computers be intelligent/conscious?
Course Objectives	Upon the completion of the course the students will be able : ▪ To understand the meaning of the term 'Artificial Intelligence', and know how the Turing test can be used to evaluate intelligent behavior ▪ To be familiar with some common searching techniques used in game-playing ▪ To know what is meant by an 'Expert System', and to have had practical experience of developing a simple expert system using Prolog ▪ To understand the operation of Single and Multi-Layer Perceptrons and Hopfield networks, and to have had practical experience of programming a simple neural network in C ▪ To be aware of some of the philosophical issues raised by the field of Artificial Intelligence

Course Content

- Introduction
  - What is Artificial Intelligence?
  - History of AI
  - Evaluating 'intelligent' behavior: the Turing test
- Game playing
- Expert Systems
  - What are expert systems

    - What have they been used for?
    - Programming a simple expert system in Prolog
- Uncertainty
    - Reasoning with uncertainty
    - Probabilistic reasoning
- Machine learning
- Overview of neural networks
- Philosophical issues
    - What is intelligence?
    - What is consciousness?
    - Computers vs. Humans
    - Can computers be intelligent/conscious?
        - The 'Chinese Room'

Delivery Method Lecture, Laboratory

Course Requirements

- Project work (60%),
- Final Exam (40%)

References

- "Artificial Intelligence – A Modern Approach",
  S. Russell & P. Norvig,
  Prentice-Hall, 2003.
- "Neural Networks – A Comprehensive Foundation",
  S. Haykin,
  Prentice-Hall, 1994.
- "From Logic Programming To Prolog",
  K. Apt,
  Prentice-Hall, 1996.

Course Title	**Complexity Theory**
Course Code	COMP321
Credit/Contact Hours	3 Credit Hrs(3 Lec Hrs)
Year and Semester	3rd Year 2nd Semester
Course Category	Major
Pre-requisite(s)	▪ COMP321 – Formal Language Theory
Rationale	This course is about problems and computational complexity of solutions.
Course Description	Turing Machine(TM): Standard TM, Computability of TM, Construction of TMs; Computability: undesirability problems: halting problem, some unsolvable problems about TMs, some unsolvable problems about CFLs; Computational complexity: Big-O notation, non-deterministic polynomial time, class P vs class NP, polynomial time reduction and NP-complete problems, Cook's theorem; Random walks. Counting classes; Unique solutions. Interactive proofs
Course Objectives	Upon the completion of this course the students will be able to:- ▪ Have a clear concept about TM ▪ Identify recursion function and enumerable languages ▪ Identify undesirability problems ▪ Understand computational complexity ▪ Understand non-uniform (circuit) complexity ▪ Have a clear concept about random walks

Course Content

- Introduction to P and NP
  - Introduction
  - Review of Turing machines, P, and NP.
  - Time/space hierarchy theorems.
  - NP completeness and reductions.
- NP completeness
  - More on NP-completeness.
  - Self-reducibility.
  - Ladner's theorem
- Deterministic and non-deterministic space complexity
  - On co-non-deterministic languages.
  - NL and NL-completeness.
  - Savitch's theorem;

- The Immerman-Szelepcsenyi theorem.
- Non-uniform (circuit) complexity
  - Introduction to circuit complexity
  - P/poly
- More on circuit complexity. Randomized computation
  - The Lupanov bound on circuit size.
  - Randomized complexity classes: RP, coRP, BPP.
  - Error reduction using independent and pairwise-independent sample spaces.
- The polynomial hierarchy
  - Mahaney's theorem.
  - The polynomial hierarchy.
  - The Karp-Lipton theorem
- More on randomized computation
  - ZPP and PP.
  - The relation of BPP to P/poly and the polynomial hierarchy.
  - Randomized space classes; random walks on undirected graphs.
- Random walks. Counting classes
  - Markov chains and random walks on undirected graphs.
  - #P and #P-completeness.
  - Approximate counting
- Unique solutions. Interactive proofs
  - The Valiant-Vazirani theorem and hardness of finding unique solutions Interactive proofs.
  - AM, MA and their relation to IP.
  - Round reduction for interactive proofs.
- The PCP theorem and applications to hardness of approximation
  - Introduction to PCP; applications to hardness of approximation.
  - NP &sube PCP(poly, O(1)).

Delivery Method Lecture, Student-lead Presentation

Course Requirements

- Presentation (30%)
- Mid Exam (30%)
- Final Exam (40%)

References

- O. Goldreich," Introduction to Complexity Theory",(July 31, 1999).
- S. Homer and A.L. Selman," Computability and Complexity Theory.", Springer, 2001.

Course Title	**Computer Graphics**
Course Code	COMP378
Credit/Contact Hours	4 Credit Hrs(3 Lec Hrs and 2 Lab Hrs)
Year and Semester	3rd Year 2nd Semester
Course Category	Major
Pre-requisite(s)	▪ COMP216 – Data Structures and Algorithms ▪ COMP225 – Object-Oriented Programming ▪ COMP124 – Fundamentals of Programming II ▪ COMP123 – Fundamentals of Programming I
Rationale	Many applications rely on graphical input for modeling or analysis purposes. This course acquaintance students with basic principles and tools for creating graphical elements; ranging from primitive to 3D.
Course Description	Introduction to computer graphics: applications of computer graphics; graphics input/output devices; overview of OpenGL/glut; simple graphics primitives; coordinate transformations; homogeneous projections; windows and viewports in OpenGL; clipping, zooming, panning & tilting; OpenGL viewing pipeline; polygonal meshes; vectors; scene rendering: depth cueing, OpenGL virtual camera, shading models, lighting and reflections, hidden surface removal, texture mapping, rendering techniques
Course Objectives	Upon the completion of the course the students will be able : ▪ To have a good understanding of the role and importance of graphics in computer systems ▪ To know how to draw simple objects using C++/Open GL, and how to model more complex objects using simple primitives ▪ To be familiar with managing windows and viewports in Open GL ▪ To understand the mathematics underlying object transformations and projections, including the use of homogeneous coordinates ▪ To be familiar with scene-rendering in C++/Open GL, and the aware of the techniques which can be used to produce photo-realistic rendering

Course Content

- Introduction
  - Applications of computer graphics
  - Graphics input/output devices
  - Introduction to Open GL/Glut
- Simple graphics primitives

    - Line drawing
    - Polygon drawing
    - Interaction with mouse/keyboard in OpenGL
- Coordinate transformations
    - Homogeneous coordinates
    - 2D/3D transformations
    - Rotations, translations, scalings
    - Perspective and orthographic projections
- Windows and Viewports in Open GL
    - What are windows and viewports?
    - Managing windows and viewports
    - Clipping
        - Finding intersections between lines/planes
    - Zooming, panning and tilting
    - Open GL viewing pipeline
        - Modelview transformation
        - Viewport transformation
        - Projection transformation
- Representing objects with polygonal meshes
- Vectors:
    - Dot and cross products
    - Surface normals
- 3-D graphics & rendering
    - The virtual camera in Open GL
        - View volumes
    - Lighting & reflections
        - Ambient light
        - Diffuse & specular reflections
    - Shading models
        - Gouraud
        - Phong
    - Depth cueing
        - Fogging
        - Perspective projections
    - Hidden surface removal
        - Using surface normals
        - Z-buffering
    - Texture mapping
    - Rendering techniques
        - Ray-Tracing
        - Radiosity rendering
        - Surface rendering vs volume rendering

Delivery Method Lecture, Laboratory

Course Requirements

- Project – 60%
- Final Exam – 40%

References

- "Computer Graphics Using Open GL",
  F. S. Hill,
  Prentice-Hall, 2001.
- "Computer Graphics, C Version",
  D. Hearn & M. P. Baker,
  Prentice-Hall, 1997.

Course Title	**Advanced Programming**
Course Code	COMP328
Credit/Contact Hours	4 Credit Hrs(3 Lec Hrs and 2 Lab Hrs)
Year and Semester	3rd Year 2nd Semester
Course Category	Major
Pre-requisite(s)	▪ COMP261 – Fundamentals of Database Systems ▪ COMP225 – Object-Oriented Programming ▪ COMP124 – Fundamentals of Programming II ▪ COMP123 – Fundamentals of Programming I
Rationale	Enterprise level application software can be said to have Data source, Domain logic, and presentation constituent parts. This course deals with these aspects and allows students create a fully-blown application.
Course Description	Java assertions, collection classes, searching and sorting, regular expressions, logging, bit manipulation, serialization, threads, networking with sockets, Remote Method Invocation, and Java Database Connectivity, and Java Graphical user Interface.
Course Objectives	Upon the completion of this course the students will be able to:- ▪ Understand and apply concurrent programming ▪ Design and implement user interface ▪ Write a program that can connect to a specific database ▪ Implement localization ▪ Write network programming

Course Content

- Review of Java Fundamentals
  - The Java Environment
  - Data Types
  - Strings
  - StringBuffers
  - Arrays
  - Passing Data Types to a Method
  - Constructors and Initialization
  - Inheritance
  - Abstract Classes
  - Interfaces
  - Static Data, Methods and Blocks
  - Wrapper Classes
- Packaging a Java Application

    - Introduction
    - Packages
    - Managing Source and Class Files
    - jar Files
    - The Manifest File
    - The javadoc Utility
    - Documenting Classes and Interfaces
    - Documenting Fields
    - Documenting Constructors and Methods
- Miscellaneous Enhancements.
    - Enhanced for Loop
    - Autoboxing and Auto-Unboxing
    - Static imports
    - Varargs
    - Typesafe Enums
    - Formatted Strings
    - Format Specifier Syntax
    - Format Specifier Conversions
    - Format Specifier Flags
    - Formatted Integers
    - Formatted Floating Points
    - Formatted Dates
    - Formatted Complex
- Assertions
    - Assertion Syntax
    - Compiling with Assertions
    - Enabling and Disabling Assertions
    - Assertion Usage
- Regular Expressions
    - Regular Expressions
    - String Literals
    - Character Classes
    - Quantifiers
    - Capturing Groups and Back references
    - Boundary Matchers
    - Pattern and Matcher
- The Java Collection Classes
    - Introduction
    - The Arrays Class
    - Searching and Sorting Arrays of Primitives
    - Sorting Arrays of Objects
    - The Comparable and Comparator Interfaces
    - Sorting - Using Comparable
    - Sorting - Using Comparator
    - Collections
    - Lists and Sets
    - Iterators
    - Lists and Iterators Example
    - Maps
    - Maps and Iterators Example
    - The Collections Class

    - Rules of Thumb
- Generics
    - Introduction
    - Defining Simple Generics
    - Generics and Subtyping
    - Wildcards
    - Bounded Wildcards
    - Generic Methods
- Advanced I/O
    - Introduction
    - Basic File I/O Example
    - Buffered I/O
    - The Console Class
    - Object Serialization
    - Serialization Issues
    - Compressed Files
    - Writing Your Own I/O Classes
    - Property Files
    - The Preferences Class
- Enhanced I/O
    - Introduction
    - Channels
    - Buffers
    - Typed Buffers
    - Direct Buffers
- Logging API
    - Introduction
    - Loggers
    - Logger Levels
    - Logger Handlers
    - Specifying Handlers and Formatters
    - Configuring Handlers
    - LogManager
- Networking
    - Networking Fundamentals
    - The Client/Server Model
    - InetAddress
    - URLs
    - Sockets
    - A Time-of-Day Client
    - Writing Servers
- Threads and Concurrency
    - Introduction
    - Creating Threads by Extending Thread
    - Creating Threads by Implementing Runnable
    - Advantages of Using Threads
    - Daemon Threads
    - Thread States
    - Thread Problems
    - Synchronization
    - Performance Issues

- Networking Fundamentals
    - The Client/Server Model
    - Internet Addresses
    - URLs
    - Sockets
    - Sample Client Programs
    - Writing Servers
- Java Database Connectivity (JDBC)
    - Introduction
    - Relational Databases
    - Structured Query Language
    - Transactions
    - Meta Data
- Java Graphical User Interface
    - Layout managers
        - Layout Managers
        - FlowLayout Manager
        - Managing Your Own Layout
        - BorderLayout Manager
        - GridLayout Manager
        - Combining Layout Managers
        - CardLayout Manager
    - Components
        - Introduction
        - The AWT Hierarchy
        - Graphics context
        - Buttons
        - Labels
        - Text Fields
        - Text Areas
        - Panels
        - Canvases
        - Checkboxes
        - Choices
        - Lists
        - Scrollbars
        - Menus
    - Event handling
        - The Event Hierarchy
        - Event Classes
        - Listener Interfaces
        - Button Events
        - Text Events
        - Focus Events
        - Item Events (CheckBox, Choice, and List)
        - Key Events
        - Mouse Events
        - Window Events
        - Scrollbar Events
        - Other Events
        - Event Handling Styles

    - The swing components
        - Introduction to Swing
        - Model View Controller Architecture
        - JFrame
        - JApplet
        - JButton
        - JLabel
        - JTextField
        - JTextArea
        - JScrollBar, JScrollable, JScrollPane, JViewPort
        - JList
        - JMenuBar, JMenu, JMenuItem
        - Pluggable Look-and-Feel
- Java Application Deployments
- Localization
    - Locales

Delivery Method Lecture, Laboratory

Course Requirements
- Project – 60%
- Final Exam – 40%

References
- Harvey M. Deitel, Paul J. Deitel, Sean E. Santry, "***Advanced Java(***TM) 2 Platform How to ***Program***"

Course Title	**Introduction to Neural Networks**
Course Code	COMP358
Credit/Contact Hours	3 Credit Hrs(2 Lec Hrs and 2 Lab Hrs)
Year and Semester	3rd Year 2nd Semester
Course Category	Major
Pre-requisite(s)	▪ COMP357 – Introduction to Artificial Intelligence ▪ COMP216 – Data Structures and Algorithms ▪ COMP225 – Object-Oriented Programming ▪ COMP124 – Fundamentals of Programming II ▪ COMP123 – Fundamentals of Programming I
Rationale	This course enables students understand artificial neural networks and learning algorithms.
Course Description	History of neural computing; biological neurons; models of neurons; activation functions; overview of artificial neural networks (ANNs); AI and ANNs; supervised and unsupervised learning; learning algorithms; perceptrons & learning; Hopfield networks; Boltzmann machines; Kohonen networks; coding and representation; memory, biological plausibility of ANNs; applications of ANNs
Course Objectives	Upon the completion of the course the students will be able to: ▪ Understand the basic operation of biological neuronal networks ▪ Be aware of the different types of artificial neural network, and have had practical experience of implementing a number of them in C ▪ Understand how information can be represented using a neural network ▪ Understand how neural networks can learn and remember

Course Content

- Introduction
  - History of neural computing
- Neurons
  - Biological neurons
  - The organization of neurons in the brain
  - Models of neurons
    - Types of activation function
- Artificial Neural Networks (ANNs)
  - Terminology

- Feedback
- AI and ANNs

- Learning
  - Supervised vs. unsupervised learning
  - Error-correction learning
  - Hebbian learning
  - Competitive learning
  - Boltzmann learning
- Perceptrons
  - Single Layer
  - Multi-Layer
  - Learning: back-propagation
- Other ANN architectures
    - Hopfield networks
    - Boltzmann machines
    - Kohonen networks
- Coding and Representation
  - How information is represented in neural networks
  - Distributed representations
- Memory
  - Theories of memory
- Biological plausibility of ANNs
- Applications of ANNs

Delivery Method Lecture, Laboratory

Course Requirements

- Mid Exam (30%)
- Assignment (30%)
- Final Exam (40%)

References

- "Neural Networks – A Comprehensive Foundation", S. Haykin, Prentice-Hall, 1994.

Course Title	**Internet Programming**
Course Code	COMP439
Credit/Contact Hours	4 Credit Hrs(3 Lec Hrs and 2 Lab Hrs)
Year and Semester	4th Year 1st Semester
Course Category	Major
Pre-requisite(s)	▪ COMP132 - Computer Networking and Data Communication ▪ COMP328 – Advanced Programming ▪ COMP225 - Object-Oriented Programming
Rationale	The internet has proved to be the information highway for entertainment, commerce, even governance. This course enables students the associated protocols, the web, HTML, and Java Server Side technologies(servlet and JSP).
Course Description	Introduction to the Internet; Internet terminology; brief history of the Internet; overview of web development tools; HTTP/HTTPS; HTML; Internet protocols; XML; Applets, servlets, and JSP, web security in Java; databases on the Internet; Web security; cryptography theory; e-commerce; web design
Course Objectives	Upon the completion of the course the students will be able to: ▪ To understand the basic operation of the Internet and be familiar with Internet terminology ▪ To be aware of the different ways of developing web sites ▪ To understand how HTTP allows reliable communication between computers on the WWW ▪ To be aware of some important Internet communications protocols ▪ To know how to develop web pages using HTML ▪ To understand the XML markup language and be familiar with a number of extended markup languages based on XML ▪ To be familiar with using the FrontPage web site editor ▪ To understand what is meant by an Applet, JSP and a Servlet, and know how to implement Applets and Servlets in Java ▪ To be aware of the security issues relating to Java Applets and Servlets ▪ To be aware of some security issues of the Internet and know some techniques for addressing these concerns ▪ To be familiar with the basics of cryptography theory, and be aware of the security infrastructure in place on the Internet ▪ To be familiar with the different techniques for maintaining state between web page requests ▪ To be familiar with the concept of e-commerce, and understand

some of the issues involved in developing e-commerce web-sites
- To have completed a short web site development project

Course Content

- Introduction
  - What is the Internet?
  - Terminology
  - Brief historical development
- Overview of web development tools
  - HTML/XML
  - Scripting: client-side vs. server-side
    - CGI/JSP
  - Java applets/servlets
  - Editors: FrontPage/Dreamweaver/MS Visual InterDev/Text editors
- HTTP/HTTPS protocols
  - The HTTP request Model
  - The HTTP response Model
  - Status codes
  - Secure websites
- Other Internet protocols
  - TCP, SMTP, POP, FTP
- HTML
  - HTML sections and tags
  - Text formatting
  - Lists
  - Comments
  - Tables
  - Frames
  - Hyperlinks
  - Graphics
  - Cascading style sheets
- XML
- Web servers
  - MS Internet Information Server
  - Others – Apache, Netscape
- Dynamic web content creation
  - Applets
  - The Servlet technology
    - Cookies
    - Session tracking
  - JSP
    - Introduction
    - JSP tags
      - Declaration
      - Expression
      - Directive
      - Scriplet
      - Action
    - JSP implicit object

        - JSP Session tracking
    - Combining Servlet and JSP
        - Model/View/Controller
- Data-Centric web applications
- Web security
    - Security risks of web scripting
    - Defensive programming
    - Denial of Service attacks
    - Viruses and worms
- Cryptography theory
    - Encryption
    - Secret key cryptography & public key cryptography
    - Key agreement protocols
    - Digital signatures
    - Public key infrastructure
    - Security protocols
- Overview of e-commerce programming
- Principles of good web design

Delivery Method Lecture, Laboratory

Course Requirements

- Project Work (60%)
- Final Exam (40%)

References

- "Weaving a Website: Programming in HTML, Java Script, Perl and Java", S. Andersen-Freed, Prentice-Hall, 2002.
- "XML In A Nutshell", E. R. Harold & W. S. Means, O'Reilly Publishing, 2002.
- "The Complete E-Business and E-Commerce Programming Training Course – Student Edition", H. M. Deitel, P. J. Deitel & T. Nieto, Prentice-Hall, 2001.
- Java in a Nutshell, David Flanagan, O'Reilly, 2002

Course Title	**Professional Ethics in Computer Science**
Course Code	Comp401
Credit/Contact Hours	2 Credit Hr (2 Lecture)
Year and Semester	4th year 1st semester
Course Category	Major
Pre-requisite(s)	*None*
Rationale	This course is about the computer professional ethical codes.
Course Description	historical perspectives of the relationship between society and technology; the information society and the information revolution ;knowledge and information economies ;technological determinism and social choice ;access and inequality; censorship and desirability of unfettered access ;control and surveillance ;the virtual self and society; professional aspects of computing :intellectual property, privacy, computer crime and ethics
Course Objectives	Upon the completion of this course the students will be able to:- ▪ Understand the relationship between information and society ▪ Know the effect of computer technology in business and society ▪ Know the issues in wireless networks ▪ Know what ethical codes a computer professional should have

Course Content

- Introduction
  - Key concepts
  - Historical perspectives of the relationship between society and technology
- The information society and information revolution
- Knowledge and information economies
  - Information society and the effect of the information revolution on work
  - Work from pre-industrial revolution through to teleworking
- Technology Determinism and Social choice
  - Running controversy as to how far technology does,or does not ,condition social change
- Access and inequality
  - Reality behind the idea of the global village put forward by those who argue that the world is in the process of an information revolution
- Censorship and desirability of unfettered access
  - Access to information and the relationship between access and power

- Problems raised by pornography and terrorism
- Control and Surveillance
  - Control and surveillance in relation to the use of various forms of information technology
  - Electrical panopticon and the information mosaic
  - Balancing the right of the individual against the rights of the collective
- The virtual self and society
  - The concept of self and identity raised by social interaction in virtual environments
- Professional aspects of computing
  - Intellectual property
  - Privacy
  - Computer crime and ethics

Delivery Method Lecture

Course Requirements
- Mid Exam (40%),
- Final (60%)

References
- C.Beardon &D.WhiteHouse,Ablex "computers and society", 1994

Course Title	**Computer Hardware and Maintenance**
Course Code	COMP437
Credit/Contact Hours	1 Credit Hr(1 Lecture Hr and 1 Lab Hr)
Year and Semester	4th Year 1st Semester
Course Category	Major
Pre-requisite(s)	▪ COMP233 - Computer Architecture and Organization ▪ COMP234 – Operating Systems
Rationale	This course focuses on Personal Computer hardware and software aspects. Covered will be, maintenance, precaution measures.
Course Description	PC hardware: power supply, ports and connectors, motherboards, video systems, sound systems, modems, input devices, memory, processors; boot process; BIOS; buses; data recovery techniques; PC troubleshooting
Course Objectives	Upon the completion of the course the students will be able : ▪ To have a good understanding of the operation of the internal components of a PC ▪ To understand the boot process and BIOS of a PC ▪ To understand the importance of data recovery techniques, and know how to recover backed up data ▪ To have had practical experience of troubleshooting PC hardware problems

Course Content

- PC hardware
  - The enclosures (case or chassis)
  - Power supplies
  - Ports and connectors
  - Motherboards
  - Video systems
  - Sound systems
  - Drive systems
  - Modems
  - Input devices
  - Standardized form factor
  - Memory
  - Processors
- The PC boot process
  - The PC hierarchy

        - Hardware
        - BIOS
        - Operating systems
        - Application
    - MS-DOS system files
    - The boot process
- BIOS
    - Typical motherboard BIOS
    - BIOS features
- Buses
    - Industry Standard Architecture (ISA)
    - Peripheral Component Interconnect (PCI)
    - Accelerated Graphics Port (AGP)
- Data recovery techniques
    - Understanding data loss
    - Recovery files and folders
    - Recovery FAT and directory damage
    - Recovering the MBR (Master Boot Record)

Delivery Method Lecture, Laboratory

Course Requirements

- Lab Exercise – 50%
- Final Exam – 50%

References

- "Mike Meyers A+ Certification Passport",
  M. Meyers & T. Rosenblath,
  Osborne, 2002.
- "How Computers Work",
  R. White & T. Downs,
  MacMillan Computer Publishing, 2001

Course Title	**Analysis of Algorithms**
Course Code	Comp417
Credit/Contact Hours	3 Credit Hr (3 Lecture Hrs)
Year and Semester	4th Year 1st Semester
Course Category	Major
Pre-requisite(s)	COMP216 - Data structure and algorithms
Rationale	This course is about analysis and design of algorithm correctness and performance.
Course Description	Review of the basic data structure, Measuring Complexity, Advanced Searching & Sorting Algorithms, Hashing algorithm, Design techniques (Divide and conquer ,Dynamic programming, greedy algorithm, Graph algorithm), Elementary Graph algorithm (Breadth first search, Depth first search, strongly connected components, minimum spanning tree, shortest paths), String Matching, Computational geometry)
Course Objectives	Upon the completion this course the students will be able to:- ▪ Apply advanced searching and sorting algorithms ▪ Develop, and reason about the correctness and performance of algorithms, in particular for string searching and graph manipulation ▪ Have clear understanding of design and analysis of algorithm

Course Content

- Basic Principles of Algorithm Design and Analysis
- Data Structures:
  - Stacks,
  - queues
  - linked lists
  - trees
  - binary search trees
  - heaps
  - graphs,
  - sets
  - union-find
- The Divide and Conquer Method:
  - Overall technique
  - merge sort

- quicksort
- quickselect
- FFT, etc.

- The Greedy Method
  - Overall technique,
  - the knapsack problem
  - optimal merge pattern
  - Huffman coding
  - minimum spanning tree
  - single-sourse shortest paths problem, etc
- Dynamic Programming:
  - Overall technique
  - matrix chain problem
  - all-pairs shortest path problem
  - optimal binary search trees, etc.
- Graph Traversal Techniques:
  - Tree traversal and applications,
  - depth-dirst search,
  - bread-first search,
  - connectivity algorithms
  - biconnectivity algorithms, etc.
- Backtracking:
  - Overall technique
  - generation of combinatorial objects such as graphs
  - sets
  - permutations
  - graph colorings
  - cliques
  - Hamiltonian cycles, etc.
- Branch and Bound method:
  - Overall method
  - the 0/1 knapsack problem,
  - the job assignment problem
  - the traveling salesman problem, etc.
- Lower bound theory
  - Techniques for determining complexity lower bounds of problems
  - algorithm modeling
  - application to lower bound on sorting
  - searching, and merging.
- Introduction to the Theory of NP-completeness:
  - Nondeterministic algorithms,
  - complexity classes
  - NP-completeness,
  - problem reduction,
  - Specific NP-complete problems

Delivery Method Lecture

Course Requirements

- Mid Exam (30%)
- Assignment (30%),
- Final Exam (40%)

References

- Cormen, leiserson and Rivest, “Introduction to Algorithms”, McGraw-Hill, 2001

Course Title	**Computer Security**
Course Code	Comp438
Credit/Contact Hours	2 Credit Hr (2 Lecture)
Year and Semester	4th year 2nd semester
Course Category	Major
Pre-requisite(s)	▪ COMP337 - Systems Programming ▪ COMP 234 - Operating systems ▪ COMP 233 - Computer Architecture and Organization ▪ COMP 132 - Computer Networking and Data Communication
Rationale	This course is about the elements of Computer Security and its management.
Course Description	Introduction and overview of computer security: purpose, elements of Computer security; Roles and responsibilities, common threats; management controls: Program Policy**,** Structure of a Computer Security Program.
Course Objectives	Upon the completion of the course the students will be able to: ▪ Understand the overview of computer security ▪ Know the Roles and responsibilities of computer security ▪ Know the management controls of computer security

Course Content

- Introduction
  - Purpose
  - Intended Audience
  - Organization
  - Important Terminology
  - Legal Foundation for Federal Computer Security Programs
- Elements of computer security
  - Computer Security Supports the Mission of the Organization. Computer Security is an Integral Element of Sound
  - Management.
  - Computer Security Should Be Cost-Effective
  - Computer Security Responsibilities and Accountability Should
  - Be Made Explicit
  - Systems Owners Have Security Responsibilities Outside Their
  - Own Organizations

    - Computer Security Requires a Comprehensive and Integrated
    - Approach.
    - Computer Security Should Be Periodically Reassessed.
    - Computer Security is Constrained by Societal Factors
- Roles and responsibilities
    - Senior Management
    - Computer Security Management
    - Program and Functional Managers/Application Owners Technology Providers
    - Supporting Functions
    - Users
- Common threats: a brief overview
    - Errors and Omissions
    - Fraud and Theft
    - Employee Sabotage
    - Loss of Physical and Infrastructure Support
    - Malicious Hackers
    - Industrial Espionage
    - Malicious Code
    - Foreign Government Espionage
    - Threats to Personal Privacy
- Computer security policy
    - Program Policy
    - ssue-Specific Policy .
    - System-Specific Policy
    - Interdependencies
    - Cost Considerations
- Computer security program management
    - Structure of a Computer Security Program
    - Central Computer Security Programs
    - Elements of an Effective Central Computer Security
    - System-Level Computer Security Programs
    - Elements of Effective System-Level Programs
    - Central and System-Level Program Interactions
    - Interdependencies
    - Cost Considerations
- Computer security risk management
    - Risk Assessment
    - Risk Mitigation
    - Uncertainty Analysis
    - Interdependencies
    - Cost Considerations
- Security and planning in the computer system life cycle
    - Computer Security Act Issues for Federal Systems
    - Benefits of Integrating Security in the Computer System Life Cycle
    - Overview of the Computer System Life Cycle
    - Security Activities in the Computer System Life Cycle
    - Interdependencies
    - Cost Considerations
- Assurance
    - Accreditation and Assurance
    - Planning and Assurance

- Design and Implementation Assurance
- Operational Assurance
- Interdependencies
- Cost Considerations

- Personnel/user issues
  - Staffing
  - User Administration
  - Contractor Access Considerations
  - Public Access Considerations
  - Interdependencies
  - Cost Considerations

Delivery Method Lecture

Course Requirements

- Mid Exam (40%),
- Final (60%)

References

- C.Beardon &D.WhiteHouse,Ablex "computers and society", 1994
- The NIST Handbook ,"An Introduction to Computer Security" ,Special Publication 800-12

Course Title	**Information Storage and Retrieval**
Course Code	COMP466
Credit/Contact Hours	3 Credit Hrs(3 Lec Hrs)
Year and Semester	4th Year 2nd Semester
Course Category	Major
Pre-requisite(s)	▪ COMP216 – Data Structures and Algorithms
Rationale	This course is given to acquaint students with modern tools and techniques for information and storage and retrieval.
Course Description	Concepts in information Storage and retrieval, basic theories and principles of information storage and retrieval; content analysis, vocabulary control and indexing techniques automatic Indexing and Searching, modern Information Retrieval Models
Course Objectives	Upon the completion of the course the students will be able to: ▪ Understand the processes of information storage and retrieval; ▪ Understand modern information retrieval models; ▪ Design, develop and evaluate Information Retrieval systems; ▪ Understand evaluation issues in IR;

Course Content

- Introduction to information storage and retrieval;
- Foundations, basic theories and principles of information storage and retrieval;
- Content analysis;
- Vocabulary control and indexing techniques;
- Automatic Indexing and Searching;
- Modern Information Retrieval Models;
    - Boolean Model
    - Vector space model
    - Probabilistic models
- Data/File Structures and Algorithms in IR;
- Query Analysis and Search Strategies;
- Retrieval Evaluation;
- User Interfaces in IR;
- Current Research Issues in IR.

Delivery Method Lecture

Course Requirements

- Mid Exam – 40%
- Final Exam – 60%

References

- R. Baeza-Yates, B. Ribiero-Neto. (1999). Modern Information Retrieval, ACM Press, Addison-Wesley: New York
- R. R. Korfhage. (1997). Information Storage and Retrieval, John Wiley & Sons: New York C.J. van Rijsbergen (1979). Information Retrieval. London: Buterworths
- G. Salton and M.J. McGill. (1983). Introduction to Information Retrieval. McGraw-Hill: New York
- K. Sparck Jones and P. Willett (1997). Readings in Information Retrieval. Morgan Kaufmann: San Francisco
- I.H. Witten, A. Moffat and T.C. Bell (1999). Managing gigabytes: Compressing and Indexing Documents and Images. Morgan Kaufmann: San Francisco

Course Title	**Software Engineering**
Course Code	COMP428
Credit/Contact Hours	4 Credit Hrs(3 Lec Hrs and 2 Lab Hrs)
Year and Semester	4th Year 1st Semester
Course Category	Major
Pre-requisite(s)	▪ COMP216 – Data Structures and Algorithms ▪ COMP225 – Object-Oriented Programming
Rationale	Software Development needs staffing, risk management, processes, among other things. This course acquaintances students with such principles.
Course Description	Classical software engineering; formal specification of software requirements; software cost estimation; safety-critical systems & risk assessment; software testing; software maintenance; fundamentals of object-orientation; UML notation
Course Objectives	Upon the completion of the course the students will be able to: ▪ To understand the importance of the stages in the software lifecycle ▪ To understand the UML notation ▪ To be aware of modern development methodologies ▪ To understand the issues involved in large-scale software development ▪ To have had practical experience of software engineering using Java

Course Content

- Classical software engineering
  - The software design process
  - Software process models:
    - Waterfall model
    - Prototyping model
    - Boehm's spiral model
    - Concurrent development model
- Formal specification of software requirements
- Software cost estimation
  - Programmer productivity
  - Algorithmic cost modeling
  - COCOMO model
- Safety-critical systems & risk assessment

- Software testing & maintenance
- Fundamentals of object-orientation
    - Objects, classes, encapsulation, inheritance, polymorphism, aggregation, abstract classes.
    - Process and techniques of OO analysis and design.
- UML notation

Delivery Method Lecture, Student-lead presentation

Course Requirements

- Project work (60%)
- Final Exam (40%)

References

- "Software Engineering: A Practitioner's Approach",
  R. Pressman,
  McGraw-Hill, 2001.
- "The Unified Modelling Language User Guide",
  G. Booch, J. Rumbauch, I Jacobson,
  Addison-Wesley, 1999.
- "Software Engineering",
  I. Sommerville,
  Addison-Wesley, 1996

Course Title	**Compiler Design**
Course Code	COMP426
Credit/Contact Hours	4 Credit Hours(3 Lec Hrs and 2 Lab Hrs)
Year and Semester	4th Year 2nd Semester
Course Category	Major
Pre-requisite(s)	▪ COMP233 - Computer Architecture and Organization ▪ COMP124 - Fundamentals of Programming II ▪ COMP123 – Fundamentals of Programming I ▪ COMP216 - Data Structures and Algorithms ▪ COMP225 - Object-Oriented Programming
Rationale	This course acquaintances with principles and tools for designing compilers.
Course Description	Introduction to compilers; compilers and interpreters; historical development; structure of a compiler; phases and passes; lexical scanning; types of grammar; backtracking; syntax analysis: Backus Naur Form, top-down and bottom-up parsing; code generation
Course Objectives	Upon the completion of the course the students will be able to: ▪ To have a good understanding of the operation of compilers ▪ To understand how Backus Naur Form can be used to describe language syntax ▪ To know what is meant by 'lexical scanning', and be familiar with some of the common techniques employed in lexical scanners ▪ To be familiar with code generation techniques for some common programming language constructs

Course Content

- Introduction
  - What is a compiler?
  - Compilers and interpreters
  - Historical development of compilers
  - The structure of a compiler
  - Phases and passes
- Lexical scanning
  - What is lexical scanning?
  - Types of grammar
    - Unrestricted

        - Context sensitive
        - Context free
        - Regular
    - Backtracking
- Syntax analysis
    - Describing a language's syntax – Backus Naur Form
    - Top-down parsing
    - Bottom-up parsing
- Code generation
    - Arithmetic expressions
    - Program statements and control structures
- Optimization
- Case study – implementing a simple compiler

Delivery Method Lecture, Laboratory

Course Requirements

- Project (60%)
- Final Exam (40%)

References

- "Building Your Own Compiler With C++",
  J. Holmes,
  Prentice-Hall, 1995.
- "The Art of Compiler Design – Theory and Practice",
  T. Pittman & J. Peters,
  Prentice-Hall, 1992.

Course Title	**Expert Systems**
Course Code	Comp597
Credit/Contact Hours	3 Credit Hr (2 Lecture Hrs,2 Lab Hrs)
Year and Semester	
Course Category	Major Elective
Pre-requisite(s)	*None*
Rationale	Basic principles for building expert systems are discuss. The representation of knowledge for use by expert systems for decision making is explained.
Course Description	Brief history of expert systems; why expert systems, economic reasons, time, space, consistency, quality in decision making, intellectual reasons, human cognitive shortcomings. Pure reasoning systems vs. knowledge rich systems. Knowledge acquisition; meaning, purpose and techniques. Knowledge representation; frames, rules, classes and procedures. An introduction to expert system development tools and techniques. Inference methods; forward and backward chaining, depth and breadth approaches, rule selection strategies. Explanation; how/why, symbolic and non-symbolic systems, probability/certainty factor, case based reasoning.
Course Objectives	Upon the completion of this course the students will be able to:- ▪ Understand the basic principles of Expert Systems and expert systems design ▪ Understand the various reasoning techniques ▪ Use the different tools of expert system design to represent simple rules ▪ Select and suggest appropriate reasoning method for a given problem

Course Content

- Brief history of expert systems
- why expert systems
  - economic reasons
  - time
  - space,
  - consistency,
- quality in decision making,
  - intellectual reasons,
  - human cognitive shortcomings.

- Pure reasoning systems vs. knowledge rich systems.
- Knowledge acquisition;
    - Meaning
    - purpose and techniques.
- Knowledge representation
    - frames,
    - rules,
    - classes and procedures.
- An introduction to expert system development tools and techniques.
- Inference methods;
    - forward and backward chaining,
    - depth and breadth approaches,
    - rule selection strategies.
- Explanation;
    - how/why,
    - symbolic and non-symbolic systems,
    - probability/certainty factor,
    - case based reasoning.

Delivery Method Lecture, Lab

Course Requirements

- Mid Exam (30%),
- Assignment (30%),
- Final (40%)

References

- Patterson, "*Introduction to AI and Expert Systems"*
- Giarratano and Riley, "*Expert Systems: Principles and Programming"*

Course Title	**Mobile Computing**
Course Code	Comp595
Credit/Contact Hours	3 Credit Hr (2 Lecture Hrs,2 Lab Hrs)
Year and Semester	
Course Category	Major Elective
Pre-requisite(s)	*None*
Rationale	This course acquaintance students with how wireless networks and their associated technologies work.
Course Description	Basics of wireless propagation: fading, path loss, antennas, modulation; variety of wireless networks: PAN, LAN, WAN, Sensor networks, RFIDs; Issues in wireless networks: multi-access, mobility, low-power, security, wless errors; Measurements in WLAN networks; Wireless mesh networks; WPANs; Sensor networks; Basics of cryptography and security; Overview of the WiMax (802.16) system/standard ;GSM system overview; Authentication and Privacy in mobile systems; Mobile-IP;TCP over wireless
Course Objectives	Upon the completion of this course the students will be able to:- ▪ Understand the basic principles of wireless propagation ▪ Understand variety of wireless networks ▪ Know the issues in wireless networks

Course Content

- Introduction
    - Basics of wireless propagation
    - fading
    - path loss
    - antennas,
    - modulation
- variety of wireless networks:
    - PAN, LAN, WAN, Sensor networks, RFIDs;
    - Issues in wireless networks: multi-access, mobility, low-power, security, wless errors
- Measurements in WLAN networks
- Wireless mesh networks
- WPANs
- Sensor networks
- Basics of cryptography and security

- Overview of the WiMax (802.16) system/standard
- GSM system overview
- Authentication and Privacy in mobile systems
- Mobile-IP
- TCP over wireless

Delivery Method Lecture, Lab

Course Requirements

- Mid Exam (30%),
- Assignment (30%),
- Final (40%)

References

- Theodore S. Rappaport, Pearson Education, "Wireless Communications: Principles and Practice", 2nd Edition

Course Title	**Visual Programming**
Course Code	COMP593
Credit/Contact Hours	3 Credit Hrs(2 Lec Hrs and 2 Lab Hrs)
Year and Semester	
Course Category	Major Elective
Pre-requisite(s)	
Rationale	Rapid Application Development for less risks problem domains has recently gained popularity among developer. This course enables students get acquainted with such a programming language.
Course Description	Introduction to visual programming; events and event procedures; designing the user interface; adding controls; menus & dialogue boxes; visual basic programming: simple statements, Arrays, control structures, procedures
Course Objectives	Upon the completion of the course the students will be able to: ▪ To understand what is meant by a "visual" programming language ▪ To have good knowledge of the Visual BASIC programming environment ▪ To have practical experience of developing applications with Visual BASIC

Course Content

- Introduction
  - What is Visual BASIC?
  - Events and event procedures
  - Developing applications in Visual BASIC
- Getting started:
  - The Visual BASIC environment
  - Creating a new project
  - Saving and opening projects
  - Running projects
  - Getting help
- Designing the user interface
  - The Project window
  - The Form window
    - Adding controls

                - Displaying output data (labels & text boxes)
                - Entering input data (text boxes)
                - Selecting multiple features (check boxes)
                - Selecting exclusive alternatives (option buttons & frames)
                - Selecting from a list (list boxes & combo boxes)
            - Menus & dialogue boxes
                - Adding drop-down menus
                - Submenus
                - Adding pop-up menus
                - Adding dialogue boxes
        - Modifying object properties
        - Code editor window
- Visual BASIC programming
    - Simple statements
        - Variables and constants
        - Arithmetic operators
            - Operator precedence
        - Strings
        - Assignment statements
        - Input and output statements
        - Comments
    - Arrays
        - Declaring array variables
        - Accessing array elements
    - Control structures
        - Conditional expressions
        - If-Then statements
        - If-Then-Else statements
        - Select-Case statements
        - For-Next loops
        - Do loops
        - While loops
    - Procedures
        - Subroutines
        - Functions

Delivery Method Lecture, Laboratory

Course Requirements

- Project – 60%
- Final Exam – 40%

References

- "Visual Basic",
  B. S. Gottfried,
  Schaum's Outlines, 2001.
- "The Complete Visual Basic 6 Training Course – Student Edition",
  H. M. Deitel & P. J. Deitel,
  Prentice-Hall, 1999.

Course Title	**Computer Vision and Image Processing**
Course Code	COMP591
Credit/Contact Hours	3 Credit Hrs(2 Lec Hrs and 2 Lab Hrs)
Year and Semester	
Course Category	Major Elective
Pre-requisite(s)	
Rationale	This course acquaintance students with mathematical principles and image processing tools for manipulating images for enhancement, extracting meaning, and so on.
Course Description	Introduction to image processing; image enhancement in spatial/frequency domain; image restoration; colour image processing; image compression; image segmentation; object recognition
Course Objectives	Upon the completion of the course the students will be able to: ▪ To be aware of the range of applications for image-processing algorithms ▪ To understand how images can be represented in the frequency domain, and to be familiar with the mathematics of the Fourier transform ▪ To understand how convolution filters can be used in image restoration and feature detection ▪ To be familiar with some basic image compression techniques, and be aware of the common compression formats used for images To be familiar with the basics of image segmentation and object recognition

Course Content

- Introduction
  - Terminology
  - Applications of image-processing
  - Pixels and images
  - Areas of image processing
  - The human visual system
  - Digital image fundamentals
- Image enhancement
  - Spatial domain vs. frequency domain methods
  - Spatial domain

        - Point processing
            - Grey-level transformation functions
                - Image negatives
                - Log transformations
                - Power-law transformations (gamma correction)
                - Piecewise-linear transformations
                - Grey-level slicing
            - Histograms/histogram equalisation
            - Arithmetic/logic operations
                - NOT, AND, OR, subtraction, addition, averaging
        - Mask processing/filtering
            - Basics of spatial filtering/convolution
            - Smoothing filters
                - Linear smoothing filters
                - Median filter
            - Sharpening filters
                - Laplacian
                - Unsharp masking
                - Gradient filters
    - Frequency domain
        - Fourier transform/frequency domain/frequency components
        - Filtering in the frequency domain
            - Notch filter/lowpass filter/highpass filter
            - Correspondence with spatial domain filtering
        - Lowpass filters
            - Ideal
            - Butterworth
        - Highpass filters
            - Ideal
            - Butterworth
- Image Restoration
    - Model of the degradation process
    - Noise models
        - Spatial and frequency properties of noise
        - Important noise PDFs
            - Gaussian/Rayleigh/uniform/impulse
            - Periodic noise
            - Estimation of noise parameters
    - Spatial filtering for restoration
        - Mean filters
            - Arithmetic
            - Geometric
            - Harmonic
            - Contraharmonic
        - Order-statistics filters
            - Median
            - Maximum
            - Minimum
            - Midpoint

        - Adaptive filters
    - Frequency domain filtering for restoration
        - Bandreject filters (ideal, Butterworth)
        - Bandpass filters (ideal, Butterworth)
        - Notch filters (ideal, Butterworth)
    - Geometric transformations
- Colour Image Processing
    - Colour fundamentals
    - Colour models
        - RGB model
        - CMY/CMYK models
        - HSI model
    - Pseudocolour image processing
        - Intensity slicing
        - Grey-level to colour transformations
    - Full-colour image processing
        - Colour transformations
            - Colour complements
            - Colour slicing
            - Tone and colour corrections
            - Histogram processing
        - Colour smoothing
        - Colour sharpening
        - Colour segmentation
            - Segmentation in HSI colour space
            - Segmentation in RGB colour space
        - Colour edge detection
- Image Compression
    - Lossy vs. lossless compression
    - Fundamentals
        - Coding redundancy
            - Variable length encoding (Huffman coding)
        - Interpixel redundancy
            - Run-length encoding
        - Psychovisual redundancy
            - Grey-level quantisation and IGS
    - Image compression model
        - Source encoder and decoder
        - Channel encoder and decoder
            - Parity/Hamming codes
- Image Segmentation
    - Detection of discontinuities
        - Points, lines, edges
        - Gradient operators (Roberts, Prewitt, Sobel)
    - Edge-linking and boundary detection
        - Local processing
        - Hough transform
- Object Recognition:
    - Patterns and pattern classes
    - Recognition based on decision-theoretic methods: (p698)

- Minimum distance classifier
- Correlation

Delivery Method Lecture, Laboratory

Course Requirements

- Mid Exam – 25%
- Lab Exercise – 15%
- Project – 30%
- Final Exam – 30%

References

- "Digital Image Processing", R. C. Gonzales & R. E. Woods, Prentice-Hall.
- "Computer Vision – A Modern Approach", D. A. Forsythe & J. Ponce. Prentice-Hall.

9.2. Description of Courses – Supportive

Course Title	**Digital Electronics**
Course Code	PHYS131
Credit/Contact Hours	3 Credit Hours
Year and Semester	1st Year 1st Semester
Course Category	Supportive
Pre-requisite(s)	
Rationale	

Course Description

Digital circuits and digital techniques are widely used in computers, automation, science and technology, transportation, entertainment, space exploration, and on and on. The course explores the fundamental principles, concepts and operations that are common to all digital systems from the simplest on/off switch to the most complex computer. It focuses on how digital systems work and extends to the application and analysis of digital systems. It starts by introducing some underlying concepts that are vital part of digital technology and these concepts are expanded. It includes number systems and codes logic gates and Boolean, algebra, simplification of combinational logic, flip flops and digital arithmetic.

Course Objectives

Upon the completion of this course the students will be able to:-

- Distinguish between analog and digital representations
- Recognize the basic characteristic of the binary number system
- Convert a number from one number system to its equivalent another number system.
- Use Boolean algebra and the Karnaugl, map as tools to simplify logic circuits.
- Analyze or design digital logic circuits based on traditional techniques using gates flip flops and memories.
- Make use of Boolean algebra in the analysis and design of digital circuits.
- Construct and analyze the operation of different flip flops.
- Sketch the output wave forms of several flip flops in response to a set of input signals.
- Perform computations of digital arithmetic.
- Describe and outline more modern methods of design.

Course Content

- Number Systems and Codes
  - Digital and Analog systems
  - Binary to decimal conversion
  - Decimal to binary conversion
  - Octal number system
  - Hexadecimal number system
  - BCD code
- Logic gates and Boolean algebra
  - Boolean constants and variables
  - Logic gates and truth tables
  - OR operation with OR gates
  - AND operation with AND gates
  - NOT operation
  - Boolean algebra and digital circuits
  - Boolean theorems and properties
  - DeMorgan's Theorems
  - Implementations of logic functions
  - NOR and NAND gates-as universal gated
  - XOR and XNOR gates
- Simplification of Combinational Logic
  - Sum of products and product of sums form
  - Simplifying logic circuits
  - Algebraic Simplification
  - Designing combinational logic circuits
  - Kamaugh map method
  - XOR and XNOR gates
  - "Don't care" values
- Flip-Flops and Related devices
  - NAND gate latch
  - NOR gate latch
  - Clocked S-C Flip-Flop
  - Clocked J-K Flip-Flop
  - Clocked D Flip-Flop
  - D-Latch
  - Master/slave Flip-Flops
  - Flip-Flop applications
  - Flip-Flop synchronization
  - Data storage and transfer
  - Shift registers
- Operations and Circuits
  - Binary addition
  - Representing signed numbers
  - Binary arithmetic signed numbers (2's complement), addition, subtraction, multiplication, division
  - BCD addition
  - Hexadecimal Arithmetic
  - Arithmetic circuits
  - Circuits for binary addition: Half adder and Full adder

Delivery Method Lecture

Course Requirements

- Home take assignments( )
- Mid-Exam( )
- Final-Exam( )

References

- ALBERT PAUL MALVLNO-ELECTRONIC PRINCIPLES
- TAUB AND SHILLING-DIGITAL INTEGRATED ELECTRONICS
- RONALD J. TOCCI, NEAL S. WIDMER-DIGITAL SYSTEMS

Course Title	**Applied Mathematics I**
Course Code	MATH141
Credit/Contact Hours	4 Credit Hours(4 Lec hours and 2 Tutorial Hours)
Year and Semester	1st Year 1st Semester
Course Category	Supportive
Pre-requisite(s)	
Rationale	
Course Description	Vector and vectors in space, matrices and determinants, Limits and Continuity; Derivatives; Application of the derivative; Function; Inverse of a function and its derivative, inverse trigonometric, hyperbolic functions and their derivatives; L'Hopital's rule; Integration; Techniques of integration (by parts, substitution, partial fraction, trigonometric integrals, trigonometric substitution); application of integration (Volume, arc length, surface area); Improper Integrals.
Course Objectives	Upon the completion of this course the students will be able to:-

- Describe vectors and vector spaces.
- Identify the operations on vectors.
- Determine equations of lines and planes in space.
- Define a matrix, observe the operations on matrices.
- Determine the inverse of a matrix (if it exists).
- Find the determinant of a square matrix.
- Describe the different properties of determinants.
- Apply a determinant to find area, Volume of certain geometrical figures.
- Compute limits of various functions; check the continuity of functions
- Use the concept of limit to define derivative
- Compute derivatives of various functions
- Apply differential calculus to solve a real life problems.
- Use the concept of limits to define a definite integral
- Compute definite and indefinite integrals
- Appreciate the applications of differential and integral calculus to other fields of sciences.

Course Content

- Vector space, Matrices and Determinants
  - Vectors and Vector Space

        - Cartesian coordinates in space
        - Vectors in space
        - The dot product
        - The cross product & triple products
        - Lines is space
        - Planes in space
    - Matrices
        - Operations with matrices
        - Properties of matrix operations
        - The inverse of a matrix
    - Determinants
        - the determinant of a matrix
        - Evaluation of a determinant using elementary operations
        - Properties of determinants
        - Applications of determinants
- Limits and Continuity
    - Limit, definition and examples
    - Limit Theorems
    - Different Types of Limits
    - Continuity
    - Intermediate Value Theorem (IVT).
- Derivatives
    - The derivative
    - Differentiability
    - The chain Rule
    - Derivatives of sum, product and quotient, trigonometric, logarithmic,
    - Hyperbolic functions and their inverses
    - Higher derivatives
    - Implicit differentiation
    - L'Hopital's rule
- Application of the Derivatives
    - Relate rates
    - Mean Value Theorem (MVT) and its application
    - Derivative Tests (1st and 2nd tests)
    - Extrema of a function and its application
    - Curve sketching
    - Tangent line approximation and Differentials
- The Integral
    - The definite Integral
    - Properties of the definite integral
    - Fundamental Theorems of calculus
    - Change of Variables
    - Indefinite Integrals
    - Integration rule (techniques)
        - Integration by substitution
        - Integration by parts
        - Integration by trigonometric substitution
        - Integration by partial fractions
        - Application of the integral

Delivery Method

- Conducting Lecture for those chapters that need clarification and elaboration.
- Arranging Tutorial class 2 hours per week for each of the courses and discuss
- Solve problems in each work sheet provided.
- Some chapters / sections that are appropriate for Discussion will be presented in group discussion form.
- Project (if appropriate) will be given
- Some chapters / sections may be given as a Reading Assignment.
- In all the methods followed it will be tried to make it participatory.

Course Requirements

- Assignment (a minimum of two)
- Test (a minimum of two)
- Mid Exam
- Final Exam

References

-

Course Title	**Introduction to Probability and Statistics**
Course Code	MATH143
Credit/Contact Hours	4 Credit Hours(4 Lec)
Year and Semester	1st Year 1st Semester
Course Category	Supportive
Pre-requisite(s)	
Rationale	
Course Description	
Course Objectives	Upon the completion of this course the students will be able to:- ▪
Course Content	
	•
Delivery Method	Lecture
Course Requirements	▪
References	▪

Course Title	**Applied Mathematics II**
Course Code	MATH144
Credit/Contact Hours	4 Credit Hours(4 Lec hours and 2 Tutorial Hours)
Year and Semester	1st Year 2nd Semester
Course Category	Supportive
Pre-requisite(s)	MATH141- Applied Mathematics I
Rationale	
Course Description	Real Sequences; Real Series; Power and Tailor Series; Differential calculus of functions of several variables; Multiple Integrals.

Course Objectives

At the end of the course students will be able to:

- Compute a sequence.
- Use the concept of sequence to define a series.
- Determine convergence or divergence of series by using different tests of
- Convergence.
- Compute radius and interval of convergence.
- Compute Taylor series of different functions.
- Describe concept of limit & continuity and derivative of functions of several
- Variables.
- Compute directional derivatives of a function of several variables.
- Apply the concept of relative extremum to find the extrema of functions of two variables.
- Extend to concept of integration to functions of two / three variables.
- Compute double and triple integrals of functions of two/three variables.
- Compute triple integrals on spherical and cylindrical coordinates.
- Appreciate the application of multiple integral in other fields of science.
- Apply multiple integral to solve real life problems.

Course Content

- Real Sequences
  - Definitions and Notations
  - Bounded and Monotonic Sequences
  - Convergence of Sequences

- Real Series
  - Definition & examples of a series
  - Positive series
  - convergence tests of positive series
    - Integral test & Comparison test
    - Ratio test & root test
  - Alternating series
  - Generalized convergence tests
- Power Series
  - Radius of Convergence
  - Algebraic Operations Between Converging Power Series
  - Differentiation and Integration of power series
  - Taylor series & Macluarin series
  - Taylor's Formula, Applications
- Differential Calculus of Functions of Several Variables
  - Definition, Examples, Graphs
  - Limit and Continuity of a function of several variables
  - Partial Derivatives
  - Directional Derivatives and Gradient
  - Normal vector and Tangent Plane
  - Tangent plane approximations & total differential
  - Applications: Error Estimation, Approximation of Values of Function
  - Chain Rule, Implicit Differentiation
  - Relative Extreme of Functions of Two Variables
  - Largest and Smallest Values of a Function on a Given Set
  - Extreme Values under Constraint Conditions, Lagrange's Multiplier
- Multiple Integrals
  - Double Integrals and Their Calculation by Iterated Integrals
  - Double Integrals in Polar Coordinates
  - Application: Area
  - Triple Integrals and Their Calculation by Iterated Integrals
  - Triple Integrals in Spherical and in Cylindrical Coordinates
  - Application: Volume, Centre of Mass

Delivery Method

- Conducting **Lecture** for those chapters that need clarification and elaboration.
- Arranging **Tutorial** class 2 hours per week for each of the courses and discuss
- Solve problems in each work sheet provided.
- Some chapters / sections that are appropriate for **Discussion** will be presented in group discussion form.
- **Project** (if appropriate) will be given
- Some chapters / sections may be given as a **Reading Assignment.**
- In all the methods followed it will be tried to make it **participatory.**

Course Requirements

- Assignment (a minimum of two)
- Test (a minimum of two)
- Mid Exam
- Final Exam

References

Course Title	**Discrete Mathematics and Combinatorics**
Course Code	MATH240
Credit/Contact Hours	3 Credit Hours(3 Lec hours)
Year and Semester	2nd Year 1st Semester
Course Category	Supportive
Pre-requisite(s)	
Rationale	
Course Description	This course includes elementary counting principles; Addition and multiplication principle; Binomial Theorem; Elementary probability theorem; Conditional probability; Random variable & expectation Recurrence relations & the techniques to solve this recurrence relations; Elements of graph theory and application of graphs.
Course Objectives	Upon the completion of this course the students will be able to:- ▪ Use the methods and principles in combinatiorics to provide answers to counting problems that arise in the study of computer science. ▪ Get acquainted with elementary ideas of probability. ▪ Use graph theory to solve network oriented problems.

Course Content

- **Combinatorics**
  - **Elementary counting principles**
    - Addition Principle
    - The inclusion-exclusion principle
    - Multiplication Principle
    - Permutations and Combinations
    - The Binomial theorem
  - **Elementary Probability Theory**
    - Sample space and events
    - Probability of an event
    - Conditional probability
    - Independent events
    - Random variables and expectation
  - **Recurrence Relations**
    - Introduction
    - Linear recurrence relations with constant coefficient
    - Solution of linear recurrence relations

- Solutions of inhomogeneous recurrence relations

- **Graph Theory**
  - **Elements of Graph Theory**
    - Introduction
    - Basic terminologies in a graph vertices, edges, order, degree of a vertex
    - Isomorphic graphs
    - Paths and connectivity
    - Complete, regular and bipartite graphs
    - Eulerian and Hamiltonian graphs
    - Tree graphs
    - Planar graphs
    - Graph Coloring
  - **Directed graphs**
    - Introduction on digraphs
    - Basic terminologies
    - Paths and connectivity
    - Rooted trees
  - **Matrices and Graphs**
    - Adjacency Matrices
    - Incidence Matrices
  - **Applications of graphs**
-

Delivery Method

- Conducting **Lecture** for those chapters that need clarification and elaboration.
- Arranging **Tutorial** class 2 hours per week for each of the courses and discuss
- Solve problems in each work sheet provided.
- Some chapters / sections that are appropriate for **Discussion** will be presented in group discussion form.
- **Project** (if appropriate) will be given
- Some chapters / sections may be given as a **Reading Assignment.**
- In all the methods followed it will be tried to make it **participatory.**

Course Requirements

- Assignment (a minimum of two)
- Test (a minimum of two)
- Mid Exam
- Final Exam

References

- Shiflet, A. B: Discrete Mathematics for Computer Science
- Lui, C. L**.** Elements of Discrete Mathematics
- S. Lipschutz (Schaum's Outline series): Discrete Mathematics
- Albertson, M.O & Hutchinson, J.P: Discrete Mathematics with Algorithms
- Mattson, H.F. Discrete Mathematics with Applications
- Galati R. Bodh; Finite Mathematics: An Introduction

Course Title	**Entrepreneurship and Small Business Management**
Course Code	MGT202
Credit/Contact Hours	3 Credit Hours
Year and Semester	2nd Year 2nd Semester
Course Category	Supportive
Pre-requisite(s)	None

Rationale

The graduates of the business management program, in addition to assuming managerial positions in all public and private organizations they are supposed to be job creators and employ themselves. The student graduate is not expected to be just a job seeker rather set up his own small business and run it successfully. This course it intentionally designed to equip students with the basic entrepreneurial skills that develops the initiation amd motivation in setting up his/her own small business firm manage it effectively and help him self and the society and avoid the sense of dependency to be hired by other organizations. The course International Marketing aims:

i. to familiarize with the fundamental skills of entrepreneurship and small business management, provide the basic concepts of entrepreneurship and its significance and contribution to an individual and the economy in general
ii. to prepare students for the life of an entrepreneur, set up and run successfully his own business rather than passively seeking job to be employed by other business firms. To this end it attempts to equip the students with the basic methods and procedures of preparing a business plan that enables to consciously set up and run a business venture

Course Description

This course will attempt initially to familiarize the students with the basic concepts and scope of entrepreneurship and small business management, the terms and definitions distinguishing features of entrepreneurs, the success factors for entrepreneurs, nature and characteristics of entrepreneurship, the concept of entrepreneurship development, and the role of entrepreneurship in economic development, and the functions of entrepreneurs. It aims students understand the significance and special contribution of small business and the advantages of running a small business. It provides a practical knowledge about the possible legal forms of business firm options with their corresponding advantages and disadvantages. It provides the basic skills in setting up a small business

and the techniques of preparing a business plan. In addition it familiarizes students with the skills of managing small business resources management, marketing in small business, financing small business, the possible growth strategies for a small business and the techniques of risk identification and management

Course Objectives

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

- Grasp the concept of entrepreneurship and small business management and its contribution
- Appreciate the significance of entrepreneurship and small business for self employment
- Develop the desired level of motivation and interest to set up their own business venture and run
- Develop the desired skills needed to prepare a business plan for a business venture
- Have a good understanding on the possible options a business firm can be legally established
- Have a full knowledge about the conceptual and the bureaucratic procedures one has to follow to set up his own business firm
- Have a good understanding about the financing mechanisms to set up a business venture
- Understand the possible strategies to grow and diversify the small firm int medium and bigger firm

Specific Objectives:

The specific objectives are expected to be designed by the course instructor at each unit and lesson provision levels:

For example, at the end of each lesson students will be expected to:

- Define important terms
- Explain the importance of the specific topic in relation to setting up your own small firm
- Identify the techniques of preparing a business plan and the general managerial tasks applicable to small business venture
- Understand the paramount importance of small business in the over all economy of the country as well as in the world
- Determine the challenges an entrepreneur faces in dealing with realizing a business idea
- Identify the factors that should be considered in making decisions for having ones own business venture.

Course Content

- Introduction (The concept of entrepreneur, entrepreneurship)
- Small business as a vital component of the economy (Characteristics of small business, Advantages and disadvantages of going into small business, Economic, social and political aspects of small business)
- Legal forms of business ownership (characteristics of the various forms of business, criteria for choosing the ownership form)
- Setting a small business (basic business idea, project identification and classification, steps of setting up a small business)

- Developing a business plan
- Management of resources (operating resources, employment law, types of management control)
- Marketing in small business (The marketing perspective, the marketing mix, marketing strategies for the small firm)
- Financing small business (Financial requirements, sources of finance, lease financing and hire purchase)
- Financial and material control in small business (Accounting systems, costing products, using budgets, analysing accounts, Inventory management)
- Growth strategies for small business
- Risk and insurance

Delivery Method Lecture, make a comment on discussed cases and student works, Creating different working groups, presentations, debating, Visits to some successful business ventures, Independent assignments , Business environment analysis and discussion supported with presentations

Course Requirements

- Group assignments(20%)
- Written individual assignments(15%)
- Classroom exercises(15%)
- Theoretical final exam(50%)

References

- Hailay Gebretinsae Beyene, Entrepreneurship and Small Business Management, 2nd Edition, 2006
- Richard M Hodgets and Donald F. Kuratco, Effective Small Business Management, 7th ed., 2001, USA

Course Title **Numerical Analysis**

Course Code MATH248

Credit/Contact Hours 3 Credit Hours

Year and Semester 2nd Year 2nd Semester

Course Category Supportive

Pre-requisite(s)

Rationale

Course Description

Course Objectives Upon the completion of this course the students will be able to:-

-

Course Content

- **Numerical Methods for Solving Equations**
  - Introduction
  - Errors: Sources of errors
    - Types of errors
    - Absolute and relative errors
  - Bisection Method
  - Newton- Raphson method
  - Secant method
  - Iteration (Successive Substitution) method
- **Interpolation and Approximation**
  - Introduction
  - Lagrange Introduction
  - Lagrange interpolation
  - Finite difference Interpolation
  - Linear Interpolation
  - Parabolic Interpolation
  - Interpolation for polynomials of degree grater than two
  - Least square approximation

- **Numerical Differentiation and Integration**
  - Numerical Differentiation by Interpolation
  - Numerical differentiation by finite differences
  - Numerical integration
    - Trapezoidal method
    - Simpson's Method
- **Numerical Solutions of Differential Equation**
  - Euler method
  - Taylor Series method
  - Runge - kutta method
- **Numerical Solutions of Matrices**
  - Introduction to matrices
  - Eigen values and Eigen vectors: Vector and matrix norms
  - Direct solution method: Gaussian elimination method
  - Iteration solution methods
  - Jacobi method
  - Gauss Seidel method
-

Delivery Method Lecture

Course Requirements

References

- Introduction to Numerical Analysis – P. A. Strock.
- Numerical Methods Volkov, 1986.
- Numerical Analysis – Richard L. Burden, 1981, second edition.
- Theory and differential Equations (Schaum's out line Series frank Ayres, 1981.
- Calculus with Analytical Geometry – Robert Ellis and Denny Glick, Third Edition
- Advanced calculus – Murray R. Spiegel
- Advanced Mathematics for Engineering and Scientists – Murray R. Spiegel

Course Title **Optimization Theory**

Course Code MATH341

Credit/Contact Hours 3 Credit Hours(3 Lecture Hours)

Year and Semester 3rd Year 1st Semester

Course Category Supportive

Pre-requisite(s)

Rationale

Course Description

Course Objectives Upon the completion of this course the students will be able to:-

-

Course Content

- INTRODUCTION; BASIC CONCEPTS
  - The linear programming problem
  - Linear programming modeling and examples
  - Geometrical solutions
- LINEAR ALGEBRA, CONVEX ANALYSIS AND POLYHEDRAL
  - Vectors
  - Matrices
  - Simultaneous linear equations
  - Convex sets and convex functions
  - Polyhedral sets and polyhedral cones
  - Extreme points
  - Representation of polyhedral sets
- SIMPLEX METHOD
  - Extreme points and optimality
  - Basic feasible solutions
  - Key to the simplex method
  - Geometric motivations of the simplex method
  - Algebra of the simplex method

    - Termination; optimality and unboundedness
    - The simplex method
    - The simplex method in tableau formal
- DUALITY AND SENSITIVITY ANALYSIS
    - Formulation of the Dual problem
    - Primal-Dual relationships
    - The Dual simplex method
    - The Primal-Dual method
    - Finding an initial Dual feasible solution: The artificial constraint technique
    - Sensitivity analysis
- THE TRANSPORTATION PROBLEM
    - Definition of the transportation problem
    - Representation of a non-basic vector in terms of the basic vectors
    - The simplex method for transportation problems
- The simplex Tableau associated with a transportation Tableau

Delivery Method Lecture

Course Requirements

References

- Mokhtar S.Bazaraa and etal(1977): Linear programming and network flows; second edition.
- Sundaram R.K., A first course in optimization theary.

9.3. Description of Courses – Common

Course Title	**Sophomore English**
Course Code	ENG101
Credit/Contact Hours	3 Credit Hours
Year and Semester	1st Year 1st Semester
Course Category	Common
Pre-requisite(s)	
Rationale	To master the skill of effective communication, trainees need, among others to master paragraph writing. This can create an opportunity to learn their major area courses while they stay in the university. Moreover, it enables the learners to communicate their ideas in writing proficiently in English.
Course Description	
Course Objectives	Upon the completion of this course the students will be able to:- ▪

Course Content

- The writing process
  - The writing anxiety
  - Strategies for handling writing anxiety
    - Have a positive anxiety
    - Understand the writing context
    - Know the subject well
  - Prewriting techniques
    - Free writing
    - Listening/brainstorming/bulleting
    - Cubing
    - Clustering/mapping/webbing
    - Journalistic questions

- Outlining
- Drafting and revising

- Paragraph development
  - Parageraph structure
    - Topic sentence
    - Supporting sentences
  - Errors in writing topic sentences
- Basic elements of writing
  - Unity
  - Organization and coherence
    - Organization
    - Cohesive devices
    - Types of transitions
  - Supporting details
    - The importance of specific details
    - The importance of adequate details
  - Effective sentence skills
- Modes of paragraph development
  - Exposition
    - Defining
    - Providing examples
    - Explaining a process
    - Comparing and contrasting
    - Providing causes and effects
    - Dividing and classifying
  - Narration
  - Description
  - Persuasion and argumentation
- Paraphrasing, summarizing, and quoting
  - Lesson one: paraphrasing
  - Lesson two: summarizing
  - Lesson three: quoting
- Writing a research paper
  - Step1: selecting a topic that interests you
  - Step2: limit your topic
  - Step3: prepare a list of sources of information o your topics
  - Step4: Read about your topic and make the purpose of your paper clear
  - Step5: Plan your paper and take notes that support your purpose
  - Step6: Write the paper
  - Step7:Use an acceptable format and method of documentation in the final version of your paper
- Giving oral presentation
  - Lesson One: Planning a presentation
  - Lesson Two: Developing effective presetnation
- Business letter writing
  - Lesson One: The parts of the letter and letter styles
  - Lesson Two: The parts of the letter and letter style

Delivery Method

- Individual work
- Paragraph work
- Essay writing

- Group work
- Pyramiding
- Paragraph and essay analysis

Course Requirements

- Writing assignments (10%)
- Writing in a classroom (10%)
- Peer assessement (15%)
- Mid-exam (25%)
- Final exam (40%)

References

- Dougherty, B.N. 1985. Compo sting Choices for Writers: A Cross-Disciplinary Rhetoric. New York: McGraw-Hill Book Company.
- Ezor, E. and Lewis, J. 1984. From Paragraph to Essay: A process approach for beginning college writing. New York: McGraw-Hill Book Company.
- Flower, H.R. 1983. The Little, Brown Handbook. Boston: Little, Brown and Company. Gebherdrt, R.C., Rodrigues, D. 1989. Writing Process and Intentions. Lexington: D.C. Heath and Company.
- Guth, H.P. 1989. The Writer's Agenda: The Wadsworth Writer's Guide (Shorter Edition). California: Wordsworth Publishing Company.
- Hammond, E.R. 1985. Informative Writing. New York: McGraw-Hill Book Company.
- Hughes, D.M. Louise K.L. and Richard W.C.1985. Glencoe English 12: Composition Speech and Grammar. California: Glencoe Publishing Company .1985.
- Glencoe English 10: Composition Speech and Grammar. California: Glencoe Publishing Company. 1985. Glencoe English 11: Composition Speech and Grammar. California:
- Glencoe Publishing Company.
- Huh, C.A. 1986. Researching and Writing: An interdisciplinary approach. Belmont:
- Wadsworth Publishing Company.
- Hyland, K. 2003. Second Language Writing. Cambridge: Cambridge University Press.
- Langan, J. 1985. College Writing Skills with Readings. New York: McGraw-Hill Company, Inc.
- . 1995. English Skills with Readings. New York: McGraw-Hill, Inc. New York: McGraw- Hill Company, Inc
- Oates, L.C. et al. 1998. The Legal Writing Handbook: Analysis, research, and writing. 2nd Ed. Random House.
- Squires, L.B. and Rombauer, M.D. 1982. Legal Writing in a Nutshell. St. Paul: West Publishing Company.

Course Title	**Communicative Skills**
Course Code	ENG102
Credit/Contact Hours	3 Credit Hours
Year and Semester	1st Year 2nd Semester
Course Category	Common
Pre-requisite(s)	Eng101- Sophomore English
Rationale	This is a task-based course aiming at helping students to develop a repertoire of language skill and study skills.
Course Description	Communicative Skill I mainly centers on using English for academic purposes. In doing so, the course will help students improve their grammar, vocabulary, productive, and receptive skills. These elements will be recycled around cross-cutting and emerging issues, including gender, youth, friendship and problems, social problems(e.g. drugs, addiction, harassment, and crime), environmental issues(e.g. global warming, desertification), indigenous knowledge, poverty reduction, etc
Course Objectives	At the end of this course, students will be able to: ▪ Develop their ability to use English in spoken and written discourse ▪ Utilize a range of strategies for learning vocabulary ▪ Listen and read and understand both spoken and written texts ▪ Analyze grammar in real texts ▪ Write paragraph considering different purposes an readers

Course Content

- Culture and tradition
  - Speaking and listening
    - Gettring to know each other and finding about each other
    - Pre-course discussion starter activities
  - Reading
    - Culture and tradition
  - Vocabulary
    - Building one's vocabulary
    - Strategies of vocabulary development
    - Guessing meaning from context
  - Writing
    - The planning of effective writing(thinking, planning, drafting, revising)

        - Paragraph writing(description) incorporating the learning from reading
- Gender Issues
    - Speaking and listening
        - Finding out opinions
        - Agreeing and disagreeing; debating
    - Reading
        - Gender issues
    - Grammar
        - Reported speech
    - Vocabulary
        - Word meaning
    - Writing
        - Paragraph writing incorporating the learning from reading
- Science and agriculture
    - Speaking and listening
        - Expressing warning and offering advice
    - Reading
        - Science and agriculture
    - Grammar
        - Active and passive voice
    - Vocabulary
        - Word meaning
    - Writing
        - Process writing(with focus on sequence, clarity, …)
        - How to make/do things(e.g. making coffee, making injera, …)
- A slave witness of a slave auction
    - Speaking and listening
        - Story telling
    - Reading
        - A slave witness of a slave auction
    - Grammar
        - Tense revision- past simple and present perfect
    - vocabulary
        - words from the text
    - writing
        - developing a short narrative text
- Corruption and development
    - Speaking and listening
        - Crime and punishment(discussion)
    - Reading
        - Corruption and development
    - Grammar
        - Expressing obligation/necessity using modals
    - Vocabulary
        - Word formation
    - Writing
        - Developing a short expository text

Delivery Method

- Individual assignments(10%)
- Group assignments(10%)
- Presentations(10%)

- Mid-Exam(30%)
- Final-Exam(40%)

Course Requirements

- Teacher-led dicussions
- Individual work
- Pair work
- Group discussion
- Presentation
- Role play

References

- Longman. (1995) *English Skills with Reading.* New York: McGraw-Hill Book Company.
- Gordon, H.H. (1989) Developing College Writing: From Substance to Style. New York: St. Martin's Press.
- John, A. (196) College English 1 and 2. Addis Ababa: AAU Printing Press.
- Thome, S. (1997) Mastering Advanced English Language. Macmillan Press Ltd
- Verderber, R. 1987. Communicate. Wadsworth Publishing Company. 1985. The challenge of Effective speaking. California

Course Title	**Introduction to Logic**
Course Code	PHIL102
Credit/Contact Hours	3 Credit Hours
Year and Semester	1st Year 2nd Semester
Course Category	Common
Pre-requisite(s)	None

Rationale

Course Description This course attempts to introduce the fundamental concepts of logic and methods of logical reasoning, The purpose of the course is to develop in learners the skills required to construct sound arguments of their own and the ability to critically evaluate the arguments of others: cultivate the habits of critical thinking and develop sensitivity to the clear and accurate use of language.

Course Objectives Upon the completion of this course the students will be able to:-

- Familiarized with the fundamental concepts of logic;
- Analytically introduce the students with fundamental methods of logical reasoning;
- Develop the skills required to construct sound arguments of their own and the ability to critically evaluate the arguments of others; and
- Cultivate the habits of critical thinking and develop sensitivity to the clear and accurate use of language.

Course Content

- Introduction: What is Logic?
- The Nature of Arguments
  - What is an Argument?
  - Types of Arguments: Deductive and Inductive Arguments
  - Validity and Invalidity: Truth and Falsity
  - Sound and Unsound Arguments
  - Strength and Weakness: Truth and Falsity
  - Cogent and Uncogent Arguments
- Definitions
  - Cognitive and Emotive Meanings of Terms
  - Intension and Extension of Terms
  - Definitions and their Purposes
  - Definitional Techniques

    - Criteria for Lexical Definitions
- Informal Fallacies
    - Fallacies of Relevance
    - Fallacies of Weak Induction
    - Fallacies of Presumption
    - Fallacies of Ambiguity
    - Fallacies of Grammatical Analogy
- Syllogistic Logic
    - Categorical Propositions: Standard Form and Types
    - Squares of Opposition: Traditional and Modern
    - Rules of Immediate Inference and Formal Fallacies
    - Categorical syllogism: Standard Form, Mood and Figure
    - Syllogistic Rules and Formal Fallacies
    - Methods of Testing Validity
- Propositional Logic
    - Compound Propositions & Propositional Connectives
    - Truth Functional Connectives & the Truth Value of Propositions
    - Propositional Type Arguments and Formal Fallacies
    - Symbolizing Propositions & Propositional Arguments
    - Rules for Propositional Logic: Rules of Implication and Rules of Equivalence
    - Natural Deduction
- Induction
    - Analogy and Legal and Moral Reasoning
    - Causality and Mill's Methods
    - Hypothetical Reasoning

Delivery Method Lecture, Class Discussion, Group presentations and discussions

Course Requirements

- Individual Assignment(10%)
- Mid-Exam(20%)
- Final-Exam(70%)

References

-

Course Title	**Civics and Ethical Education**
Course Code	CIVETH200
Credit/Contact Hours	3 Credit Hours
Year and Semester	2nd Year 1st Semester
Course Category	Common
Pre-requisite(s)	None

Rationale

Course Description This course is designed to be offered as a common course to all trainees in the degree program in order to produce responsible, well informed, and competent citizens. The course encompasses the basic concepts of civics and Ethics, state and government, citizenship, the values and principles of democracy and the issues in development as well as environment and sustainable development. The course also encompasses the basic concepts of constitution and constitutionalism, fundamental human rights, conflict and conflict resolution, election, globalization and major issues of applied ethics and civic virtues.

Course Objectives Upon the completion of this course the students will be able to:-

- Explain the subject matters of civics and ethics
- Understand the concept of citizen and citizenship
- Familiarize with ethics and moral issues
- Appreciate the difference between state and government
- Practice the principles and values of democracy
- Know the concept of constitution and constitutionalism
- Understand the principles of the FDRE constitution
- Explain the basic concepts and features of human rights
- Understand and analyze the concept of election and electoral processes in Ethiopia
- Discuss major issues in applied ethics and civic virtues

Course Content

- Introduction
  - The definition of Civics and ethics
  - Scope of Civics and ethics
  - Origin and historical development of Civics and Ethics

- Sources of Civics and ethics
- The Goals, objectives and significance of civics and ethics

- Ethics and Ethical Issues
  - What is Ethics?
    - Origin and Subject matter of Ethics
    - Morality and Law
    - Ethics and Morality in the Ethiopian context
  - Ethical Issues
    - The need for Ethical Education
    - Professional Ethics
    - Developmental Ethics
    - Environmental Ethics
    - Public Service Ethics
    - 6. Civic Virtues
- Fundamentals of State and Government
  - Understanding Essentials of the State
    - Conceptual Framework: Defining State, Nation, Society and Country
    - The Origin and Development of state
    - The modern concept of state and its major Attributes (Elements Qualifications)
  - Forms of State Structure
    - Unitary State Structure
      - Conceptual Approach: Defining Unitarism
      - Essential features of unitary state structure
      - Major factors /Rationale for opting unitary state structure and the nature of society
      - Types of unitary state structure
      - A potential Advantages and disadvantages of unitary state structure
        - Federal State Structure
        - Conceptual Approach: Defining the meaning and concept of federalism
        - Key features of Federal State Structure
        - Basic Rationale /Reasons/ for Opting (Preferring choosing) federal form of state structure over Unitarism; with particular.
        - Basic of various Basis Federalism; the case of Ethiopia and USA in comparative perspective
        - Potential Merits and Demerits of federalism
        - Understanding Government
        - The conceptual Framework: Definition and Meaning of the Modern concept of Government.
        - The Relationships and differences of government and state
        - The Issue of the Legitimacy, Authority, and Sources of Authority (power) of government
        - Purposes and Functions of Government
        - Essential Issues (factors) that distinguish government from other institutions

(organizations)
        - Organization of government
            - Vertical Arrangement /Organization/
                - The Federal government
                - The Sub- National /Regional/ Governments
            - Horizontal Arrangement (Organization) – Organs of Government
                - The Legislature organ
                - The Executive organ
                - The Judiciary organ
        - Modern Systems /Forms of Government
            - Presidential System of Government
            - Parliamentary System of Government.
        - Contemporary Types (Classification) of government
- Understanding Essentials of Citizenship
    - Meaning and definition of citizenship
    - Aspects/Dimensions of citizenship
    - Ways of acquiring citizenship/nationality in Ethiopia
    - Ways of loosing citizenship/nationality in Ethiopia
    - The entitlement of citizenship/nationality both to humans and dehumanized elements
    - Main challenges and prospects of citizenship (Ethiopia focus)
- Constitution and Constitutionalism
    - Defining Constitution of State
    - Principles, Purposes and Objectives of Constitution
    - Types of Constitutions
    - Constitutionality (constitutional order)
    - Constitution and Constitutional Development in Ethiopia
        - Traditional constitution
        - The 1931 constitution
        - The 1955 revised constitution
        - The 1987 constitution
    - ETHIOPIAN CONSTITUTIONAL EXPERIENCE SINCE 1991
        - The Transitional Charter of Ethiopia
        - The 1995 F.D.R.E. Constitution
- Democracy and Human Rights
    - Understanding Democracy
        - Conceptual frame work: Meaning and definitions of democracy and democratic system.
        - Dimensions of democracy
        - Understanding Rights and Obligations as Basic Pillars Democracy.
        - Ways of exercising democracy
        - Fundamental principles and values of Democracy
        - DEMOCRATIC Rights as Enshrined in FDRE constitution
    - Understanding Human Rights
        - Meaning and Definition Human Right

        - The Contemporary Significance of Human Rights
        - States' Obligations and Human Rights
        - Instruments, INSTITUTIONS AND PROCEDURES FOR ENFORCEMENT OF HUMAN RIGHTS
        - Fundamental rights and freedoms under the FDRE constitution
        - Limitations and derogations of fundamental rights and freedoms under the FDRE Constitution
- Election and Electoral Processes in Ethiopia
    - Typology of Elections and Electoral process
        - Definition, Functions, and Ways of Political Election
        - Electoral Process
        - Electoral Systems
        - Electoral Mechanisms and Principles
        - The Procedures of Electoral Process
        - Electoral Institutions and Procedures
        - Ethical Rules, Codes and Standards
        - Voting and Its Out comes
    - Voting Behaviours
        - Meaning and Definition of Voting Behaviour
        - Factors affecting/Shapes of Voting Behaviour
        - Election: The Voter and the Roles of Media
        - The role of civic organizations in elections
        - Elections and voting behaviour of citizens
    - Election and Electoral Processes In Ethiopia: Historical and Contemporary Perspective
        - General Introduction
        - Election, Electoral Systems and processes Under the Imperial Period
        - Election, Electoral Systems and processes Derge Period
        - Election, Electoral Systems and processes post-1991 Ethiopia
- Conflict, Conflict Resolution and Citizenship
    - Defining conflict: Conceptual Approach
    - Sources, Forms of Expression and Major Levels of conflict
    - Functions of Conflict and Elements that contribute dysfunctions of conflict
    - Basic categories of conflict
    - Positive and Negative consequences of Conflict
    - Positive Attitudes towards Conflict
    - Peaceful Formal and Informal Methods of Conflict Resolution
    - Conflict Resolution Structures under the FDRE constitution
    - Domestic/Indigenous Conflict Resolution Mechanisms in Ethiopia
    -

Delivery Method Short lectures, question and answer, Project work and field work, discussion, role play

Course Requirements

- Project work (20%)
- Presentation (15%)
- mid-term exam(25%)
- Final exam (40%)

References

-

Course Title	**Introduction to Economics**
Course Code	ECON304
Credit/Contact Hours	3 Credit Hours
Year and Semester	3rd Year 2nd Semester
Course Category	Common
Pre-requisite(s)	None

Rationale

Course Description This course introduces both micro and macro economics. The following are dealt with the subject matter; definition, scope, and methods of economics; basic economic problems and the economic system, supply and demand, theory of production and costs, the profit maximizing competitive firm, review of market imperfections: monopoly, monopolistic competitive and oligopoly; national income accounting, fluctuations in economic activity, aggregate demand and supply, and policy instruments; monetary policy and fiscal policy.

Course Objectives The principal aim of this course is to introduce and acquaint the students with what economics as a subject matter deals with in general. The course is designed to discuss the micro and macro concepts in general which will be the base for the advanced courses like theories of pricing and resource allocation and aggregate economics analysis.

Course Content

- introduction
  - Definition of economics
  - Fundamental economic facts
  - The basic economic problems
  - Production possibility frontier/curve and economic growth
  - Circular flow of economic activities
  - The use of graphs and equations in economics
  - Alternative economic system
    - Free market economic system
    - Command economy
    - Mixed economy
- demand and supply
  - demand
    - Meaning of demand, demand curve and demand schedule
    - Change in demand and change in quantity demand

        - Individual and market demand

    - supply
        - Meaning of supply, supply curve and supply schedule
        - Change in supply and change in quantity supplied
        - Individual and market supply
    - elasticity of demand and supply

    - Market equilibrium
        - Equality of demand and supply curves
        - Change in equilibrium price and quantity
- consumer behavior
    - meaning of utility
    - Utility in different approaches
        - cardinal utility approach
        - ordinal utility approach
    - Utility under cardinal utility approach
        - TU, MU, LDMU
        - Equilibrium under cardinal utility approach
    - ordinal utility approach
        - Meaning of indifference curve,
        - Characteristics of indifference curve
        - Meaning of budget line
        - Characteristics of budget line
        - Equilibrium under ordinal utility approach
- theory of production
    - Meaning of production
    - Economics of production, economic and technical efficiency
    - Periods of production: short run and long run production periods
    - Equilibrium in short run period of production: TP,MP,AP,DMR and optimum stage of production
    - Long run period of production
        - Meaning and characteristics of isoquants
        - Equilibrium condition and the underlying meaning
- Theory of cost
    - Meaning of cost of production
    - Short run cost of production
        - TC,TFC,TVC
        - ATC,AFC,AVC,MC
        - basic u shaped curves
        - invertible relationship among ATC, AVC,MC,AP,MP
    - Long run cost of production
        - Meaning
        - TC,TFC,LAC and SAC
- market structure
    - perfectly competitive market
    - Monopoly
    - Monopolistically competitive
    - oligopoly
- national income accounting
    - national income accounting:

        - GDP and GNP
        - NI,NNI,PI,PDI,deperciation
    - approaches to measure national income
        - Income approach
        - Expenditure approach
        - Value added method
- business cycle
    - Meaning of business cycle
    - Why a business cycle
    - Different stages of business cycle
- fluctuations in economic activities;
    - unemployement
    - inflation
- policy instruments
    - monetary policy
    - fiscal policy

Delivery Method Lecture

Course Requirements

- Mid exam(40%)
- Final exam(60%)

References

- Bowden, Elbert v.Economics: the science of common sense 5th edition south western publishing co,1986.
- Ruffin roy j. and Gregory poul. R . Principles of micro economics 4th ed.the Christian Science publishing society, 1987.
- Armold roger. A. microeconomics, west publishing company, 1989.
- Parkin Michael. Economics. Addison, west publishing company1990
- Vaish m.c. Macroeconomics 6th ed. Vikas publishing house ltd. 1981
- Ferguson and gould Microeconomic theory. Irwin inc.1989

Course Title	**Research Methods**
Course Code	MGT401
Credit/Contact Hours	3
Year and Semester	4th Year 1st Semester
Course Category	Common
Pre-requisite(s)	

Rationale

Effective and efficient decision making lies on the availability and quality of information. Information availability again depends on research that deals with gathering, processing and interpreting data that facilitate decision making. The graduates are thus required do develop their skills of conducting research so that their decisions will be rational and logical making decision making more efficient and effective.

Course Description

This course is survey course intended to develop the student's ability in scientific analysis and to provide opportunity for application of concepts and tools in the study of socio-economic problems. Topics covered include the concepts and purposes of social research, types of research, formulating research problems and hypothesis, planning research design; collection, analysis and interpretation of data, and term project.

The course Research Methods is preceded by two courses on statistics: Basic Statistics and Managerial Statistics that deal with the fundamentals of statistics and applications of statistics in management respectively.

As a result the statistical tools and techniques required for research are not treated in detail in this course. The course assumes students have basic knowledge on the required statistical techniques. However students are required to apply appropriate statistical tools in the course, particularly, in preparing the research proposal at the end of chapter four and in conducting the mini-research at the end of the course.

Course Objectives

At the end of this course, as a general objective, students will be able to acquire basic knowledge about research methods in social science in general and in business and the public sector in particular and build the skill of conducting research in the public and government sector.

The specific objectives of the course may include the following:
At the end of the course students will be able to:

- Understand and explain the concepts, purposes and types of research.
- Determine when business research should be conducted and choose appropriate research methodology

- Formulate research problems and hypothesis
- Plan and conduct business research using different methods
- Conduct research in various fields of business and other socio–economic activities.

Course Content

- Research Methods: An Overview
- Formulation of Research Problems & Research Hypotheses
- Research Design
- Measurement and Scaling of Concepts
- Study Designs
- Data Collection
- Data Processing and Presentation
- Data Analysis
- Research Report Writing

Delivery Method

- Lecture, Using working groups, Individual assignment/ Preparing research Proposal, Group Assignment/ Conducting mini-research, Presentation (assignment papers, projects, research findings)

Course Requirements

- Written individual assignment(15%)
- Group Assignment (25%)
- Attendance and Classroom participation(5%)
- Mid semester exam(20%)
- Final exam (35%)

References

- Alreck, Pamela L. and Robert B. Settle (1985), The Survey Research Handbook, , Homewood: Irwin Inc.
- Babbie, Earl (1986), The Practice of Social Research. 4th ed. Belmont: Wadsworth , Publishing Co.
- Goode, William J. and Paul K. Hatt (1982) Methods in Social Research, Auckland: McGraw-Hill Book Company.
- Gibaldi, Joseph and Walter S. Achter (1988), MLA Handbook for Writers of Research Reports, 3rd Ed. New York: The Modern Language Association of America.
- Huit, Christine A (1986), Researching and Writing: An Interdisciplinary Approach, Belmont: Wadsworth Publishing Company.
- Leedy, Paul D (1989), Practical Research: Planning and Design, 2nd Ed. New York: Macmillan Publishing Co., Inc.: London: Coller Macmillan Publishers.
- Lin, Nan. Foundations of Social Research (1986), New York: McGraw-Hill Book Company.
- Simon, Julian L (1969) Basic Research Methods in Social Science: the Art of Empirical Investigation. New York: Random House, Inc.
- Turabian, Kate L , A Manual for Writers of Research Papers, Theses, and Dissertations. Rev.