MAKERERE UNIVERSITY

FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE

POSTGRADUATE DIPLOMA IN COMPUTER SCIENCE
programme

DAY/EVENING PROGRAMME

August 2009

Contents

1 Introduction

The Postgraduate Diploma in Computer Science aims at equiping students with advanced theoretical and practical skills in Computer Science so as to develop and manage computer systems. The objectives of the program are:

- To provide students with indepth knowledge of the theoretical and practical aspects of Computer Science so as to satisfy the technological needs in the private and public sector.
- To provide students with skills to deploy and manage Computer infrastructues in organizations so as to improve their efectiveness
- To provide students with research skills which will help then grow with the technological advancements as well as help them participate in the development of new technologies

2 Areas of Specialization

The programme has two areas of specialization. The two areas of specialization are Computer Security and Computer Vision & Image processing. A student will be required to specialize in one of them.

Th choice of the areas of specialization was dictated by the current trends in the Computing field.

2.0.1 Computer Security:

Uganda as a country, East Africa as a region and Africa as a continent is in the process of automating. Manual systems are being computerized at a fast rate and there is no evidence that this will stop soon. Computerization increases precision, speed, reliability, availability and reduces cost. Computerization has been applied in sensitive/critical areas like finance, records keeping, monitoring and tracking. Unfortunately, most organizations put emphasis on the functionality of the computerized system but pay less attention on the succeptibility of the system to be maliciously attacked by an intruder/enemy. A non secure system can be compromised and the organization that installed it registers a loss instead of a gain. In some cases, it can be thrown several years back and rendered un competitive. The program therefore aims at producing computer security experts who will be able to manage security in computerized organizations so as to maintain the competitive benefits of computerization.

2.0.2 Computer Vision and Image Processing:

Many processes in the world today can be analyzed and interpreted using their images. This can be by extracting specific features in the images and studying them. This is in a wide range of areas like medical computer vision for detection of tumors, computer vision in industries for supporting the manufacturing process and automatic detection of products for industry as well as monitoring (and controlling) of moving objects like traffic. Given the process of modernizing Uganda, high quality graduates in computer vision and image processing will add a lot of value to Ugandan/ regional organizations. Likewise, research will be able to enhance the productivity of the organizations.

The department is well placed staff wise to handle these areas of specialization. This is judged by the staff, both on ground and in training who are specializing in these areas. The Faculty of Computing and Information Technology is also in collaboration with Universities where these areas are well developed.

3 The Program and Regulations

3.1 Tuition Fees

Tuition fees for privately sponsored students shall be 3,825,000 Uganda Shillings per year for Ugandans and 3,350 US Dollars per year for international students.

3.2 Program Duration

The duration of the program is two semesters and one recess term spread in one year. Each semester has fifteen weeks of studying and two weeks of examinations. A recess term is made up of ten weeks.

3.3 Target Group

The program targets graduates from computing (Computer Science, Computer Engineering, Software Engineering, Information Technology and Information Systems) and closely related fields.

3.4 Admission Requirements

To be admitted to the Postgraduate Diploma in Computer Science, the candidate must hold an undergraduate degree in Computer Science, Computer Engineering, Software Engineering, Information Technology and Information Systems or a closely related field from a recognised institution.

3.5 Weighting

The weighting unit is the Credit Unit (CU). The Credit Unit is a series of 15 contact hours (CH) in a semester. A contact hour is equal to (i) one lecture hour (LH), (ii) two practical hours (PH) or (iii) two tutorial hours (TH).

3.6 Evaluation and Grading

Every course unit will be graded with in and at the end of the semester in which it is offered. The progressive assessment will constitute 40% of the overall mark and the final examination will constitute 60%. Grade points will be allocated to the final mark got in every course unit according to the table below:

Marks	Letter Grade	Grade Point	
90-100	A+	5	Exceptional
80- 89	A	5	Excellent
75- 79	B+	4.5	Very good
70- 74	B	4	Good
65- 69	C+	3.5	Fairly good
60- 64	C	3	Pass
55- 59	D+	2.5	Marginal fail
50- 54	D	2	Clear fail
45- 49	E	1.5	Bad fail
40- 44	E-	1	Qualified fail
0 - 39	F	0	Qualified fail

A student will be considered to have failed the course unit if he/she fails to get a grade point equal or greater than 3.

3.7 Classification

The diploma will be classified using the cumulative grade point average (CGPA)

$$CGPA = \left(\sum_{i=1}^{i=n} GP_i \times CU_i\right) \div \left(\sum_{i=1}^{i=n} CU_i\right)$$

where GP and CU represent grade points and Credit units respectively.

The diploma will be classified in accordance with the table below

CLASS	CGPA
First Class	4.40 - 5.00
Second Class - Upper Division	4.00 - 4.39
Second Class - Lower Division	3.50 - 3.99
Pass	3.00 - 3.49

3.8 Retaking of Course

A student will retake all courses where he/she has a grade point less than 3

3.9 Progression

A student can be under normal progress or probational progress.

Under normal progress, a student must have covered a full semester load, with a grade point of at least 3 in every course offered and with a cumulative grade point of at least 3.

A student who gets a cumulative grade point average of less than 3 or a grade point of less than 3 in any of the courses registerd for is under probational progress. He/she shall be allowed to progress to the next semester but shall be allowed to retake all the failed courses when they are next offered. He/she will remain on probational progress untill he/she passes it.

3.10 Progression

A student can be under normal progress or probational progress. Under normal progress, a student must have covered a full semester load, with a grade point of at least 3 in every course offered and with a cumulative grade point of at least 3. Under probational progress, a student must have a cumulative grade point average of at least 3 but with either a load lower than the semester load or at least a course with less than 3 grade points.

3.11 Discontinuation from a Program

A student will be discontinued from the program if (i) he/she is on three consecutive probartional progresses based on CGPA. (ii) He/she fails to get a grade point of at least 3 during the third assessment of a retaken course. (ii) He/she overstays o the program for more than two academic years.

4 The Curriculum

4.1 The Main Knowledge Areas

The curriculum has three main subject areas

(i) Security and

(ii) Computer Vision & Image processing.

(iii) Programming and Software development

Programming and Software Development is to be taught to all the students. Each student will chose to either specialize in Security or Computer Vision & Image processing.

4.2 The Curriculum

4.2.1 Postgraduate Diploma in Computer Science - Computer Security Option

Code	Name	LH	PH	CH	CU
Semester I (5 Courses)					
Cores:- (4 core courses)					
MCS 7118	Advanced Programming	15	60	45	3
MCN 7105	Structure & Interpretation of Computer Programs	30	30	45	3
MCN 7109	Requirements Engineering	45	–	45	3
MIT 7116	Research Methodology	30	30	45	3
Electives:- (1 elective course)					
MCS 7116	Graph Theory	45	–	45	3
MIS 7112	Modeling and Simulation	30	30	45	3
MCN 7106	Mobile Software and Content Development	30	30	45	3
	Total				15
Semester II (5 Courses)					
Core:- (4 core courses)					
MCS 7221	Security Protocols	45	–	45	3
MCS 7220	Software Security	30	30	45	3
MCS 7222	Hardware and Operating Systems Security	30	30	45	3
MCS 7226	Seminar Series	–	60	30	2
Electives:- (1 elective course)					
MCS 7215	Data Encryption	45	–	45	3
MCS 7202	Design and Analysis of Algorithm	45	–	45	3
MCS 7223	Combinatorial Optimization	30	30	45	3
MCN 7204	Mobile Applications Programming	30	30	45	3
MCN 7206	Service Oriented Architectures	30	30	45	3
MCN 7207	Software Design Process and Metrics	45	–	45	3
MIS 7202	Data Warehousing	45	–	45	3
MIT 7218	Legal and Ethical Aspects of Computing	30	30	45	3
	Total				14
Year 1: Recess Term (1 course)					
Core: - (1 core)					
PGD 6307	Postgraduate Diploma Project	–	300		5

4.2.2 Postgraduate Diploma in Computer Science - Computer Vision & Image Processing option

Code	Name	LH	PH	CH	CU
Semester I (5 Courses)					
Cores:- (4 core courses)					
MCS 7118	Advanced Programming	15	60	45	3
MCN 7105	Structure & Interpretation of Computer Programs	30	30	45	3
MCN 7109	Requirements Engineering	45	–	45	3
MIT 7116	Research Methodology	30	30	45	3
Electives:- (1 elective course)					
MCS 7116	Graph Theory	45	–	45	3
MIS 7112	Modeling and Simulation	30	30	45	3
MCN 7106	Mobile Software and Content Development	30	30	45	3
	Total				15
Semester II (5 Courses)					
Core:- (4 core courses)					
MCS 7217	Pattern Recognition	30	30	45	3
MCS 7224	Computer Vision	30	30	45	3
MCS 7225	Image Processing	30	30	45	3
MCS 7226	Seminar Series	–	60	30	2
Electives:- (1 elective course)					
MCS 7215	Data Encryption	45	–	45	3
MCS 7202	Design and Analysis of Algorithm	45	–	45	3
MCS 7223	Combinatorial Optimization	30	30	45	3
MCN 7204	Mobile Applications Programming	30	30	45	3
MCN 7206	Service Oriented Architectures	30	30	45	3
MCN 7207	Software Design Process and Metrics	45	–	45	3
MIS 7202	Data Warehousing	45	–	45	3
MIT 7218	Legal and Ethical Aspects of Computing	30	30	45	3
	Total				15
Year 1: Recess Term (1 course)					
Core: - (1 core)					
PGD 6307	Postgraduate Diploma Project		300		5

5 Detailed Curriculum

5.1 Semester I

5.1.1 MCN 7105: Structure & Interpretation of Computer Programs

(a) **Description:**
The course provides a survey of techniques and principles in the underlying design and implementation of systems. The course focuses on symbolic computation and less on numerical examples from the calculus and number theory. Effective software engineers need to know efficient techniques that serve as building blocks in the design an implementation of software systems. Today, most systems require a collection of skills to provide an efficient implementation. Therefore this course enriches through broadening rather than acceleration. The programming language used has a simple syntax and an intuitive semantic model, allowing a focus on concepts. Throughout, the focus is on understanding computational tools by building them, rather than covering many language features

(b) **Aims/Goals:**
The aim of the course is to equip students with the knowledge to understand and design software systems.

(c) **Learning outcomes**

- Strong understanding of basic concepts in computer science (including some material on lists and trees.
- Ability to write programs from scratch in the programming language Scheme while understanding the meaning of what is being written.
- Proper attention to design and testing.

(d) **Learning and teaching:**
The course is structured around a strong textbook and associated instructional development environment, though lectures offer elaboration on ideas, different examples, and additional material. Short assignments during this course are interspersed with a substantial programming project using object-oriented techniques, such as an adventure game.

(e) **Indicative Content:**

- The fundamentals of Lisp computation: names and values, evaluation, function definition and evaluation, and predicates.
- Higher-order functions, including the use of functions as parameters. Introduce the definition of functions with LAMBDA, and the use of functions that return functions as values.
- Function definition and application, making decisions (conditional expressions), working with aggregated data (structures), working with unbounded data (lists and recursion), information hiding (local definitions), functional abstraction (functions as values), mutation (changing name-value bindings), and encapsulation (making objects).
- Data abstraction and techniques for implementing "abstraction barriers. Use of scheme pairs to implement lists, trees, and other data structures. Cover advanced data abstraction techniques: tagged data, data-directed programming, and message-passing.
- State and assignment: The use of state and local assignment to write efficient programs; introduce the idea of object-oriented programming, assignment, and the environment model of evaluation that is needed to understand how local state is maintained in Scheme. Introduction to mutable data, concurrency; streams, model time-varying state information within the functional programming approach
- Meta linguistic abstraction: The creation of new programming languages, as a still more powerful abstraction technique. Two major examples are presented: Lisp and a logic programming language the course follows the fourth chapter of the text.

(f) **Reading Materials/Indicative sources:**

- Structure and Interpretation of Computer Programs by Abelson and Sussman (Second edition, MIT Press, 1996)

(g) **Assessment:**
Assessment will be by Tests and projects (40%) and Final Examination (60%)

5.1.2 MCN 7106 Mobile Software & Content Development

(a) **Description:**
This course is to give students in depth skills in the development of

mobile software and content. It covers the standards, methodology as well as practice of mobile software/content development.

(b) **Course Objective:**
The objective of the course is to teach students the principles of mobile software architecture, operating system platforms and development tools for mobile software, content and service development.

(c) **Indicative Content:**

- Introduction to Mobile Operating System Platforms (OSPs):
- Symbian, Android, Linux & Microsoft mobile.
- User Interface, Menu system, and Applications.
- Software set-up in modern Mobile Terminals.
- Overview of Mobile Multimedia Codecs.
- Fundamentals of Mobile Content & Mobile web content design.
- Widgets, W3C Standards, Device Recognition & dotMobi.
- The .mobi top level domain (TLD) initiative for mobile optimized web-site creation.

(d) **Mode of Delivery:**
The course will be delivered by means of lectures, seminars as well as laboratory exercises.

(e) **Assessment:**
Assessment will be in form of class assignments, tests and practical projects (40%) and final examination (60%)

(f) **Reading Materials/ Indicative sources:**

- Smart Phones & Next Generation Mobile Computing, P.Zheng, Morgan Kaufmann, 2006
- Mobile computing principles, R. BFar, Cambridge University Press,2005.
- http://dev.mobi. Developers guide, content developer tools.
- Developer.Symbian.com/main/getstarted,
- http://code.google.com, Android an open handset alliance project.
- Mobile Internet Architecture, Nokia IT Press-2001, ISBN 951-826-499-6.

5.1.3 MCS 7109: Requirements Engineering

(a) **Description:**
Establishing firm and precise requirements is an essential component of successful software development. This course covers a range of methods from the hard semi-formal approaches to softer methods, and some innovative techniques. Practical guidance is also included

(b) **Aims/Objectives:**
At the end of the course the student will have a breadth of knowledge about the range of requirements methods, tools, and techniques. They will gain an appreciation of at least two methods, and obtain practical guidance on elicitation techniques.

(c) **Learning outcomes:**

- To be able to clearly document non-trivial requirements
- To evaluate and formulate the user requirements in systems.

(d) **Teaching and learning patterns:** The class will be conducted on face to face, in class lecture

(e) **Indicative Content:**

- Covers the principles, tools, and techniques for requirements elicitation, specification, and analysis.
- Focus on understanding the role of requirements in system development and maintenance, and the difficulties of specifying requirements for real systems, and effective methods tools and techniques.
- This course covers the principles, tools, and techniques used to establish a software specification that captures correctly and completely the requirements of a software system under development and the expectations of the potential user.
- System and Software System Engineering, Software Requirements Concepts, Requirements Elicitation, Software Requirements Analysis, Software Requirements Specifications, Software Requirements Tools, Software Requirements Verification, Software Requirements Engineering Management, Developing a Successful Software Requirement

(f) **Reading Material/Indicative sources:**

- Requirements Engineering: Processes and Techniques (Worldwide Series in Computer Science) by Gerald Kotonya and Ian Sommerville (Aug 24, 1998)
- Software Requirements Engineering, 2nd Edition by Richard H. Thayer and Merlin Dorfman (Feb 27, 1997)

(g) **Assessment:** Assessment will be in terms of a Progressive assessment (at least 2 tests and a project) (40%) and a Final Examination (60%)

5.1.4 MIS 7112 Modeling and Simulation

(a) **Description:**
This course will introduce systems thinking, modeling, and computer simulation as a tool for the analysis, planning and management of industrial production processes and for corporate policy analysis and strategic planning. The student will obtain the knowledge and skills to conduct small simulation projects, consisting of input data analysis, model building, verification and validation, and finally interpretation of output data. Development of computer models to solve complex business problems in MIs, operations. Introduction to computer modeling techniques, discrete-event simulation and System Dynamics Modeling. Model development and testing.Aims and objectives :

(b) **Aims and objectives:**
This course provides an introduction to system modeling using both computer simulation and mathematical techniques. A wide range of case studies are examined, both in the lectures and tutorial exercises, although the emphasis is on the analysis of computer and communication systems using a combination discrete-event simulation and continuous modeling paradigms using System Dynamics Methodology.

(c) **Course Learning Outcomes:**
On completing this course students should be able to:

- Demonstrate an understanding of system modeling through the competent use of Computer Simulation methods and Mathematical Modeling techniques .
- Determine the type of systems whose behaviour can be investigated using Discrete Event Simulation and Modeling.
- Determine the type of systems whose behaviour can be investigated using System Dynamics-simulation modeling technique;

- Develop an understanding of the elements involved in the basic construction of a causal loop diagram;
- Appreciate how a verbal description of a system can be translated into a causal loop diagram and used to examine the system's behaviour;
- Translate a causal loop diagram, representing a given system, into a quantitative SD model (differential equations);
- Develop an understanding of the stages involved in the model development process.

(d) **Teaching and learning patterns:**

- Lectures, tutorial/practical sessions as well as demonstrations.
- Individual and group-based tutorial.
- Wide range of computer-based learning and other tools will be used to support the student's learning process.
- Use of real life case studies and individual literature review of current developments in the simulation and modeling

(e) **Indicative content:**

- Introduction to Simulation & Modeling with the help of relevant examples introduce concepts, uses, applications, advantages, disadvantages of simulation and modeling. Look at a detailed example of hand simulation.
- Introduction to System Dynamics Provide an introduction to system dynamics, the stages of the modeling process and develop its philosophical linkage to science and.
- Systems thinking and Causal Loop Diagramming - This section will provide an introduction to systems thinking, the use of causal loop diagrams in modeling, feedback structures and various exercises in the development of causal loop diagrams.
- Discrete Event Simulation - a practical approach to Discrete Event Simulation in the labs using CSIM software (which uses C++). Comparison with the hand simulation.
- Introduction to System archetypes, their importance and various groups present the different archetypes providing clear explanations and relevant application to different public policies.

- Stock and Flows - Introduction to differential equations (Euler and 1st Order Runge Kutta equations). A practical approach to drawing Stock and Flow Diagrams . Introduction to STELLA modeling technique. Conversion of Causal Loop Diagrams to Stock and Flow Diagrams.
- Graphical Integration Graphical integration exercises-constant rates, linearly increasing and decreasing flows, parabolic, Step functions, Ramp functions.
- Feedback Structures / Functions focus on improvement of behavior feedback dynamics. A review of different curves and other functions used in modeling such as exponentials, oscillations, S-shaped, goal seeking graphs
- Model Development A review of the phases of model development Conceptualization, formulation, testing and implementation. item Model Testing & Validation A review of the model testing and validation

(f) **Assessment method:** The assessment will be in form of tests and assignments(40%) and final written exam (60%)

(g) **Course Reference List:**

- Coyle, R. G (2001) System Dynamics Modelling : A Practical Approach; Chapman & Hall, London
- Pid, M., (1992) Computer Simulation in Management Science, 3Ed John Wiley, Chichester
- Richardson, G.P & Pugh, A L (1981); Introduction to System Dynamics Modeling with DYNAMO; MIT Press
- Steward Robinson (2004). Simulation. The Practice of Model Development and Use. John Wiley and Sons Ltd.
- Sterman, J.D. (2000). Business Dynamics : Systems Thinking and Modeling for a Complex World. Irwin McGraw Hill.

5.1.5 MCS 7116: Graph Theory

(a) **Description**
The course gives students an in depth understanding of graphs, graph algorithms and their application in a computerized setting. The applications of interest include planning, optimization and matching. The course covers the pure aspects as well as their applications specifically in computer related scenarios like routing and resource management.

(b) **Aims:**
The aims of the course are to

- Equip students with an in depth theoretical knowledge of graphs and their manipulation
- Explore ways graph algorithms can be applied in computer settings

(c) **Teaching and Learning Patterns:**
Teaching will be by class lectures and class presentations

(d) **Indicative Content**

- Graphs: Graphs and simple graphs; Graphs isomorphism; the incidence and adjacency matrices; Vertex degrees; Paths and connection; Cycles and the shortest path problem.
- Trees: Trees; Cut edges and bonds; Cut vertices; Cayleys formula and Kruskal's algorithm.
- Connectivity: Connectivity; Blocks and construction of reliable communication networks.
- Euler Tours and Hamilton Cycles: Euler tours; Hamilton cycles; The Chinese postman problem and the traveling salesman problem.
- Planar Graphs: Planar graphs; Dual graphs and Eulers formula.
- Networks: Flows; Cuts; The Max-Flow Min-Cut theorem and applications.

(e) **Assessment Method:**
Assessment will be by tests and assignment (40%) and final examination (60%)

(f) **Reading List:**

- Modern Graph Theory by Bela Bollobas, Springer 2002.
- Graph Theory: A Problem Oriented Approach by Daniel Marcus, MAA 2008.

5.1.6 MIT 7116 Research Methodology

(a) **Course Description:**
In this course, guidance will be given to students on how to identify a research problem. Instructions will be provided which will enable students to perform effective literature reviews. Students will be presented with various research paradigms and models of methodology and assist with designing an appropriate method for their research. Students will be trained in the analysis and presentation of results, exposition of processes and methods used and conclusions drawn. Guidelines outlining the preparation and writing of a research dissertation and or a project will be provided at the conclusion of the course.

(b) **Aims:**
The aims of the course are:

- To provide students with a firm foundation/underpinnings of research from which they can undertake a research problem
- To provide students with a number of separate, but related practical skills associated with the research process

(c) **Learning outcomes:**
At the end of this course unit, the students will be able to identity the aims of the research, selection of appropriate methodological approach, selection of implementation methods, data collection and analysis techniques and its interpretation, and how all this fits within the literature. In other words, the students will produce a research proposal as a blue print for the whole research dissertation and or project.

(d) **Teaching and Learning Pattern:**
Lectures will be given through out the semester. Group work and discussions to perform literature reviews will be done to enable understanding and application of concepts. This will involve identification and reading material which includes journal papers to be distributed to students a week in advance. The lecturer addresses questions to the students to encourage them to think about and understand the material. The students will identify researchable problems from which they will apply the concepts taught in class with an aim of producing research/project proposals by the end of the semester. The students will be required to build on their proposals on a weekly basis in line with the new concepts that will be taught. The students will make presentations of their draft proposal for critique and feedback from both the students and the lecturer.

(e) **Indicative Content:**
The course will cover the following topics:

- Definition of Research Methodology
- Research Paradigms in Computing and Information Systems
- Research Planning and Management
- Types of Research Methods
- Scientific writing including abstracts; identifying research problems, research objectives and questions; Interpretation of technical literature (literature reviews); Selection of overall methodological approach; Selection of suitable data collection and analysis techniques; Interpretation and conclusion of the research; and Presentation of research findings
- Research Ethics/Plagiarism

(f) **Assessment Method:**
Assessment will categorized as follows:

- Progressive assessment (40%)
  - Group work (literature reviews) 20%
  - Presentation (skills) 10%
  - Theory and application (concepts) 10%
- Exam (60%)
  - Individual work (scientific writing and research paper) 40%
  - Theory and application (concepts) 20%

(g) **Reference books:**

(i) Practical Research: Planning and Design (March 2004): Paul D. Leedy, Jeanne E. Ormrod, Jeanne Ellis Ormrod, Paperback, Prentice Hall

(ii) Graduate research: A guide for Students in the sciences (May 1998): Robert V. Smith, Paperback, University of Washington

(iii) Research Methods: A process of Inquiry ((May 2006)): Anthony M. Graziano, Michael L. Raulin, Hardcover, Prentice Hall

(iv) Introduction to qualitative research methods: A guidebook and resource (1998): Taylor, Steven J.; Bogdan, Robert, Hoboken, (3rd Ed.) NJ, US: John Wiley & Sons Inc.

5.1.7 MCS 7118: Advanced Programming

(a) **Description:**
The course gives a practical approach to programming using different languages. It is to make students explore different languages, platforms and challenges of practical importance. The course builds on the programming foundation got at undergraduate level.

(b) **Aims:**
The aims of the course are to

- Strengthen the student's programming skills
- Allow the student practically explore different programming platforms/environments
- Expose students to ways of handling important aspects of programming/software development like security, robustness, performance

(c) **Teaching and Learning Patterns:**
The course will be practical in nature. The member of staff will be able to introduce the concept in class and students will be given time to read about it and implement it on a platform determined by staff. The staff will then discuss with them any shortfalls in their implemented modules.

(d) **Indicative Content:**

- Programming for robustness
- Programming for security
- Programming for Performance
- User session management and access controls
- Programming on a chip

(e) **Assessment Method:**
Each practical assignment given to the students will contribute to the overall coursework mark (40%) and the students will sit a final written exam (60%)

(f) **Reading List:**

- Advanced Programming for the Java 2 Platform by Calvin Austin and Monica Pawlan, Addison Wesley Longman, 2000.

5.2 Semester II

5.2.1 MCS 7202: Design and Analysis of Algorithms

(a) **Description:**
The course gives students skills in development and analysis of computer algorithms. The course covers the generic approach to algorithm design, generic analysis of algorithms, studying of generalized computational problems as well as applying then to specific real life problems.

(b) **Aims:**
The aims of the course are:

- To improve the students ability to design and analysis algorithms
- To improve the problem solving skills of students
- To highlight the effect and severity of algorithm complexity on the performance of computer programs

(c) **Teaching and Learning Patterns:**
Teaching will be by lectures, class discussions and seminars

(d) **Indicative Content:**

- Design, analysis, and implementation of algorithms;
- Use of advanced data structures in algorithm design;
- Sorting,
- Algorithms for tree structures,
- Dynamic programming, greedy methods, divide and conquer
- Graph algorithms, arithmetic operations,
- Algorithms for parallel computers,
- Efficient polynomial and matrix algorithms,
- Number-theoretic algorithms, string/pattern matching,string processing and computational geometry,
- Network problems, approximation algorithms, and the theory NP completeness and approximation algorithms.

(e) **Assessment Method:**
Assessment will be by tests and Assignments (40%) and final examination (60%)

(f) **Reading List:**

- The Design and Analysis of Algorithms by Kozen, Dexter, Springer, 1992.

5.2.2 MIS 7202 Data warehousing

(a) **Course Description:**
This course includes the various factors involved in developing data warehouses and data marts: planning, design, implementation, and evaluation; review of vendor data warehouse products; cases involving contemporary implementations in business, government and industry; techniques for maximizing effectiveness through OLAP and data mining. The course in data warehousing (DW) presents the necessary fundamentals of DW (methodology, tools, techniques, systems and terminology) to students by putting these concepts into context and comparing expert views in these areas through seminars, discussions, and hands-on-work in computer labs. The prerequisite for the course is a graduate course in Database Systems before taking this course and having the skills of ER modeling, normalization, SQL and some other basic DBMS skills.

(b) **Aims:**
The main purpose of the course is to develop and gain an understanding of the principles, concepts, functions and uses of data warehouses, data modeling and data mining in business. A DW project is usually business driven and will work to improve the direction of the company by aligning the data warehouse technology with business strategy. The following areas of interest are addressed in the course:

- DW methodology
- DW architectures
- The DW development processes: Logical and physical DW
- DW data modeling
- ETL, Data access, Data quality

(c) **Learning Outcomes:**
At the completion of this unit students will have a theoretical and conceptual understanding of:

- the knowledge of theories and principles of data warehousing and OLAP;
- the potential benefits of data warehousing;

- the techniques and tools used to design a data warehouse;
- the theories and principles of data warehousing with regard to the practice of decision support; and be able to design multi-dimensional data structures;
- and appreciation of how to interact effectively with managers, consultants, and vendors in the development of a data warehouse. Upon completion of this subject, students should be able to accomplish the following:
- Demonstrate the concept of enterprise modeling as a conceptual framework in building data warehouses.
- Develop the data model using the enterprise-modeling framework, the model of the requirements for analytical functions is developed in conjunction with the development of the data model.
- Create and populate databases and develop of ETL routines, user interface, analytic applications, reports, system and application interfaces.
- Extract data from the OLTP database to the DW (they created) by addressing ETL issues.

(d) **Teaching and learning pattern:**
The course is delivered in the form of lectures, group discussions, teamwork and seminars where participants are required to actively participate both in presentation and discussions and investigate agreed upon topics.

(e) **Indicative content:**

- Requirements Analysis: The concept of analytical requirements and their differences from operational requirements are introduced. Key concepts in analytics that deal with forecasting, projection and formation of strategies are explored. Data gathering techniques such as user interviews, joint application design (JAD) are explored and practiced by doing project work in small groups.
- Conceptual Design: For the topic of conceptual design, modeling techniques specific to DW are discussed. They include the Entity-Relationship (ER) modeling and dimensional modeling including the Star Schema and Snowflake Schema, which utilize fact tables and dimension tables. Strategies in modeling with regard to the data warehouse ER model, the data warehouse dimensional model

and the independent data marts Data sourcing strategies and the logical mapping of the data schema of sourcing systems to the conceptual model can be developed during the conceptual design phase.

- Physical Design: Various levels of the physical design for the data warehouse are explored. They consist of three levels: the data level, the application level and the technical infrastructure level. Topics for the data warehouse database design include the general database design principles and specific data warehouse considerations such as de-normalization. The data level design also includes the design of data extraction, transformation and loading (ETL). At the technical infrastructure level general concepts of technical architecture for data warehousing include requirements in hardware, software and networking are discussed. The design phase also includes development of user interfaces (UI) and considerations of scalability in terms of both the growth in the number of users and the increase of use by each user.
- Development and Testing: For the topic of development, concepts and techniques in the creation of databases and applications in a development environment are introduced. They include the creation and population of databases and the development of ETL routines, user interface, analytic applications, reports, system and application interfaces. Topics of unit testing, system testing and performance testing are studied. Special topics in development sourcing strategies can be included.
- Implementation and Deployment: Different deployment strategies are discussed. They include the big-bang approach and various phased approaches. In the big-bang approach the data warehouse is deployed to the entire organization with all functionalities all at once. In the phased approach, the data warehouse can be deployed by phases based on various criteria such as geography, organizational units or data warehouse functions.
- Data Modeling: For the data modeling part of the course the starting point is to understand the basics of ER modeling and also its limitation for creating an enterprise wide data model for decision-making purposes. The STAR or dimensional modeling is used for creating data warehouses (DW). Idea behind a data warehouse is to centralize company wide information to create and deliver the necessary analytical environment, for example data mining and business intelligence; to meet the business needs. The

use of an accepted methodology provides big advantages in the conversion of the ER model to the STAR model in DW creation.

- ETL : Decision-making data are extracted from OLTP source and further organized as per fact/s or burning question/s for decision-making purposes. Data are cleansed, aggregated, transformed and loaded in the DW.

(f) **Assessment method:**
Assessment will be in terms of tests and coursework (40%) and a final examination (60%)

(g) **Reference Books:**

(i) Kimball , R., Reeves, L., Ross, M., & Thornthwaite, W. (1998 - or later editions) The Data Warehouse Lifecycle Toolkit. John Wiley & Sons

(ii) Kimball, R. & Ross, M. (2002). The Data Warehouse Toolkit : The Complete Guide to Dimensional Modelling (2nd Ed). John Wiley & Sons. ISBN: 047120024-7

(iii) Todman, C. (2001). Designing a Data Warehouse Supporting Customer Relationship Management. Upper Saddle River, NJ: Prentice Hall PTR.

5.2.3 MCN 7204 Mobile Applications Programming

(a) **Description:**
The course gives students skills in planing, designing, implementing and deploying mobile applications. issues peculiar to mobile applications like program size, robustness, etc will be covered.

(b) **Aims:**
The objective of the course is to give students a practical introduction to mobile applications software programming and development tools. The course will contain 50 % laboratory based lectures and exercises, the remaining 50 % percent is self-work and lecture seminars.

(c) **Teaching and Learning Patterns:**
Lectures, seminars and laboratory exercises

(d) **Indicative Content:**

- The dimension of Mobility.

- Principles of Mobile Applications Programming.
- Introduction to programming languages for programmable mobile phones: J2ME (MiDP), C++ (Symbian), PYTHON.
- The mobile applications and Services echo system.
- Characterization of Innovative Mobile Services and design of an application software to meet a desired criteria.

(e) **Assessment Method:**
Assessment will be in terms of progressive assessments (Tests and assignments) (40%) and final examination (60%)

(e) **Reading list:**

- http://www.Python.Org/doc, Python scripting language.
- Java ME Platform: http://java.sun.com/javame/.
- Developer.Symbian.com/main/, Symbian forum.
- http://code.google.com/android. Android an open handset alliance project.
- http://dev.mobi. Developers guide, content developer tools.
- Mobile Internet Architecture, Nokia IT Press-2001, ISBN 951-826-499-6.
- Mobile computing principles, R. BFar, Cambridge University Press,2005.

5.2.4 MCN 7206 Service Oriented Architectures

(a) **Description:**
Service Oriented Architecture (SOA) is a fairly new concept motivated by in explosive increase in services that are increasingly required to inter operate. This course provide fundamental concepts of SOA, requirements for building services, types of services such as Web services and SOA engineering as way of building dynamic, autonomous systems.

(b) **Aims:**
The course aims at giving the student an understanding of the strengths and weaknesses of a service-based architecture, informed by an ability to implement and deploy simple web services using a suitable development platform. They will also learn to define and design applications as combinations of services, and be able to discuss the emergent properties of those compositions; and to understand the research context and potential future directions for these technologies.

(c) **Learning Outcomes:**
At the end of the course, students will

- know the fundamental principle of service-oriented systems
- be able design and implement service-oriented systems

(d) **Teaching and Learning Patterns:**
The teaching method shall mainly be based on lectures. In addition, assignments such as reading and course projects shall help students to help a better understanding in the design of service-oriented systems. Furthermore, class presentations and group work will help students acquire some transferable skills.

(e) **Indicative Content**

- Software components: Modularity; reuse; contracts; component-oriented programming; services. Web-services. XML; HTTP; SOAP; WSDL; UDDI.
- Representational state transfer: Architectural styles of the web; REST; resource-oriented architecture. Composition Work flow; activity diagrams; BPMN; BPEL.
- Objects : OO middleware; CORBA; objects versus services.
- Software architecture: Client-server; layers; pipes and filters; EDA; repositories; peer-to-peer; Grid Computing.
- Semantic Web: Knowledge representation; Resource Description Framework; Web Ontology Language; Semantic Frameworks.
- Service qualities: Transactions; performance; security.
- Engineering SOA: Organization; life cycle; versioning; governance.

(f) **Reading Material/Indicative sources:**

- Building Web Services with Java: Making Sense of XML, SOAP, WSDL, and UDDI (2nd Edition) by Steve Graham, Doug Davis, Simeon Simeonov, and Glen Daniels (Jul 8, 2004)
- Web Services: Principles and Technology by Michael Papazoglou (Sep 23, 2007)
- SOA Principles of Service Design (Jul 28, 2007) by Thomas Erl

(g) **Assessment:**
Assessment will be in terms of progressive assessments (Tests and assignments) (40%) and final examination (60%)

5.2.5 MCN 7207 Software Design Process and Metrics

(a) **Description:**
Software engineers may work in groups or be required to report or measure the activities on each phase of the software engineering process. This course is a step by step description of the software metrics. It includes introduction to foundations of measurement theory, models of software engineering measurement, software products metrics, software process metrics and measuring management.

(b) **Aims:**
The course aims at giving the student an understanding and practical experience of the design metrics. The will learn and use different metrics to measure the productivity of different software systems.

(c) **Learning Outcomes:**
At the end of the course, students will

- Outline, select and apply various techniques of measuring design metrics.
- have acquired a strong grounding in different techniques of measuring software metrics.

(d) **Teaching and Learning Patterns:**
The teaching method shall mainly be based on lectures. A medium size project shall be given to students in groups or individuals. The students are expected to walk through the different software metrics.

(e) **Indicative Content:**

- Recap of SW Engineering. Goal-question-metric (GQM). Balanced Score Card (BSC).
- Requirements Metrics. Design Patterns. Design Metrics. Best Practices in Coding, Code & Test Metrics. Maintenance Metrics. Analysis Tools. Customer Satisfaction Metrics & Process Improvement Metrics. Project Management Metrics &.Human Resources & Training Metrics
- Development Quality Metrics &Quality Audit Metrics. Customer Care Metrics and Miscellaneous Metrics.

(f) **Assessment:**
Assessment will be in terms of progressive assessments (Tests and assignments) (40%) and final examination (60%)

(g) **Reading Materials:**

- Software Metrics: A Rigorous and Practical Approach, (2nd ed.) (638p.), N.E. Fenton and S.L. Pfleeger, PWS Publishing, 1998. ISBN 0-534-95425-1.
- Metrics and Models in Software Quality Engineering, Stephen H. Kan, 2nd ed. (560 p.), Addison-Wesley Professional (2002). ISBN: 0201729156.
- Software Engineering Measurement, John C. Munson, Auerbach Publications, 2003 (443 pages) ISBN:0849315034

5.2.6 MCS 7215: Data Encryption

(a) **Description:**
The course gives students skills in methods and approaches of securing data by encryption. This is as a means of preventing un authorized access by third parties. The data may be in transit or in a database/flat file. Means of protecting the data as well as the strengths and weaknesses of the different means are addressed.

(b) **Aims:**
Security of data is a very big threat to most or all computerized organizations. The course therefore aims at:

- Highlighting ways un authorized access to data can be done
- Giving an in depth understanding of ways data can be protected by encryption
- Making a comparative analysis of the strengths and weaknesses of the different cryptographic approaches

(c) **Teaching and Learning Patterns:**
Teaching will be by class lectures, seminars and class presentations

(d) **Indicative Content:**

- Conventional encryption: classical encryption techniques, modern encryption techniques, and encryption algorithms.
- Basic number theory which is used as the foundation for public-key encryption.
- Public-key cryptography such as encryption methods and digital signatures.

- Message authentication and hash functions.
- Techniques of key management, secret sharing and conducting interactive proofs.
- Comparison of encryption techniques for effectiveness.

(e) **Assessment Method:**
Assessment will be in form of tests and assignments (40%) and final examination (60%)

(f) **Reading List:**

- Data Privacy and Security: Encryption and Information Hiding by David Salomon, Springer 2003.
- Data Protection and Security for Personal Computers by Robert Schifreen, Elservier, 1992

5.2.7 MCS 7217: Pattern Recognition

(a) **Description:**
the course give students in depth understanding of the techniques and algorithm of detecting patterns and apply them in real life problems like traffic control, epidemiology and data mining

(b) **Aims:**
The aims of the course are to equip students with skills to

- Systematically analyze patterns in objects/data using the current state of the art techniques
- Use pattern recognition techniques to solve day today problems

(c) **Teaching and Learning Patterns:**
Teaching will be by lectures and class seminars as well as laboratory practicals

(d) **Indicative Content:**

- Description of patterns,
- Uncertainty in pattern recognition,
- Fuzzy sets,
- Inductive learning of rules for recognition,
- Learning discriminates,

- Self-organizing nets for pattern recognition,
- Functional link net,
- Roles of fuzzy logic,
- Pattern recognition and neural nets,
- Issues in the use of adaptive pattern recognition.

(e) **Assessment Method:**
Assessment will be by tests, assignments and laboratory projects (40%) and final written examination (60%)

(f) **Reading List:**

- Pattern Recognition by William Gibson, Putnum Adult, 2003.

5.2.8 MIT 7218: Legal and Ethical Aspects of Computing

(a) **Course Description:**
The course focuses on issues that involve computer impact on society and related concerns. The students will be taught issues on: Transitional data flow; copyright protection; Information as a source of economic power; rights to access computer systems; computer crime; data privacy; establishing national priorities in the technical and social aspects of computing; current and anticipated uses of computer prediction. The course will also examine and evaluate the meaning of ethics and professional conduct including the protection of personal ethical concerns. The students will also be exposed to the status of the regulation and emerging markets.

(b) **Aims:**
This course aims at providing students with:

- A good grounding in social, legal, ethical and management issues affecting their probable role as researchers and or working computer scientists, practitioners or engineers in Computing and Information Technology-related disciplines.
- The basic background to develop their professional role in the workplace, beyond simply performing technical tasks assigned to them.

(c) **Learning outcomes:**
Upon successful completion of this course, the students will:

- Apply the ethical concepts relevant to resolving moral issues in business, industry, and other relevant areas of concern;
- Articulate and defend with good reasons his/her own ethical point of view pertaining to specific problem areas in business, industry, and related areas;
- Analyze business plans, working procedures and policies in terms of current legislative and case law;
- Evaluate proposed and actual changes in the law for their effect on their working and personal environments in terms of rights, liabilities and responsibilities; Present compelling arguments about the social impact of new technological developments; and
- In addition, students should be able to maintain and develop their awareness of the social, legal and ethical framework in which they find themselves, through knowledge of the underlying mechanisms of change in these areas.

(d) **Teaching and Learning Pattern:**
The course will primarily be taught by external seminar speakers (i.e. professionals in the field of IT and Law related disciplines) and directed reading (from Internet resources and text books as seen in the reading list). Also interactive lectures i.e. presenting a topic to the class and giving a starting point from which the students can give their own ideas will be used in learning this course. Strong encouragement will be given for students to continue these discussions outside lectures both in person and using online discussion tools such as MUELE (Makerere University Elearning). Current IT-related legislation and case law will be taught by direct lectures, supported by directed reading. Assignments with strong formative aspects (requiring self-directed research on a topic) will support each of the sections of the course.

(e) **Indicative Content:**
The course will cover the following topics:

- Nature of ethics, ethical development, responsibilities and basic ethical directions
- Ethical principles, values, and their foundations
- Specific computing and information technology related business, industry, and engineering ethical issues
- Social impact of technological change: Internet communications; medical technologies; bio-engineering; education; entertainment;

industry, commerce and working practices; globalization; public misunderstanding of science; environmental impact of high technology

- National and international legal frameworks; specific legislation and case law involving IT issues
- Domain Names; IP law; Data Protection; Computer misuse; Software Licensing, Transitional data flow; copyright protection; Information as a source of economic power; rights to access computer systems; computer crime; data privacy; establishing national priorities in the technical and social aspects of computing

(f) **Assessment Method:**
Assessment will be made up of coursework (40%) and a final written exam (60%). Coursework will entail four parts:

(g) **Reference books:**

(i) Computer Ethics: Integrating Across the Curriculum by Marion Ben-Jacob, Mercy College, ISBN-13: 9780763778095, ISBN-10: 0763778095, Cd-rom, 2010

(ii) Pandora's Box: Social and Professional Issues of the Information Age by Andrew A. Adams and Rachel McCrindle (Paperback - 14 Dec 2007)

(iii) Engineering Ethics by Charles B. Fleddermann, 1st edition Prentice Hall, 1999. ISBN 13: 9780137842247

(iv) Engineering, Business and Professional Ethics by Moodley, Krisen, Elsevier Science & Technology 2007, ISBN-13: 9780750667418

5.2.9 MCS 7220: Software Security

(a) **Description:**
The course addresses the common security problems in softwares as well as their underlying causes. It then addresses the techniques, guidelines, principles and tools that prevent or detect them.

(b) **Aims:**
The aims of the course are:

- Explore the common security problems in software
- Explore the main causes of the security problems in software

- Explore the techniques of detecting security flaws in software
- Explore the techniques of preventing security flaws at programming level

(c) **Teaching and Learning Patterns:**
Teaching will be in terms of lectures, class presentations and practical demonstrations

(d) **Indicative Content:**

- Security threats: Buffer overflows, integer overflows, SQL injection, XSS, and race conditions.
- Techniques to prevent or detect problems: threat modeling, check lists and coding standards, static analysis tools, code reviews, typing, static analysis, language-based security (or platform-based security), security middle ware, runtime monitoring, information flow analysis, program verification, and proof-carrying code

(e) **Assessment Method:**
Assessment will be by tests and practical assignments (40%) and final examination (60%)

(f) **Reading List:**

- Building Secure Software, by John Viega and Gary McGraw. Addison-Wesley, 2002.
- The 19 Deadly Sins of Software Security, by Michael Howard, David LeBlanc and John Viega, McGraw-Hill, 2006.
- Secure Coding: Principles & Practices, by Mark G. Graff and Kenneth R. van Wyk. O'Reilly, 2003.
- Writing Secure Code, by Michael Howard and David LeBlanc, Microsoft Press, 2002.

5.2.10 MCS 7221: Security Protocols

(a) **Description:**
This course presents case studies of the design of various security protocols including SSL, WEP / WPA, IPSec, and Kerberos. We focus on discussing the pros and cons of various security trade-offs involved in the design of such protocols, and we describe vulnerabilities that some of these protocols are susceptible to due to design flaws.

(b) **Aims:**
The course aims at equipping students with systematic approaches of analysis, designing and implementing security mechanisms in organizations so as to minimize the vernulability of organization ICT infrastructure from malicious attacks.

(c) **Teaching and Learning Patterns:**
Teaching will be in terms of lectures, groupwork and class representations.

(d) **Indicative Content:**

- Fundamental concepts of Security: integrity, Confidentiality, Authenticity and Availability.
- Threat Modeling Techniques: Attack Trees, Fault Trees, etc
- Security Models: including CIA and Military model, etc
- Integrity models: including; LOCUS and Biba model, Clark-Wilson Integrity
- Logic of beliefs
- Post and Pre-Conditions
- Design of Security protocols: Requirements, Assumptions and Trusted Computing Base.
- Protocol Specification Languages: Spi-Calculus,CSP, CCP, HLPSL etc
- Analysis of Security protocols: Formal methods, proof assistants, Logic of belief, Automated tools

(e) **Assessment Method:**
Assessment will be in terms of tests and assignments (40%) and final examination (60%)

(f) **Reading List:**

- Formal correctness of security protocols by Bella Giampaolo, Springer 2007

5.2.11 MCS 7222: Hardware and Operating System Security

(a) **Description:**
This course looks at security from the perspective of the hardware and

operating system. It largely looks at smart cards which are becoming popular and they have a lot of security considerations to address. It explores common attacks on hardware/OS as well as ways such attacks can be addressed.

(b) **Aims:**
The aims of the course are to

- Equip students with knowledge security threats in computer systems caused by operating systems and hardware
- Equip students with skills of detecting such security flaws
- Explore ways such security flaws can be addressed and make the computer system more secure

(c) **Teaching and Learning Patterns:** Teaching will be in terms of lectures, class presentations and lab demonstrations

(d) **Indicative Content:**

- Smart cards: Magnetic stripe cards, smart cards, and RFID tags.
- Smart card hardware and communication.
- Contact and contact less cards.
- Memory and microprocessor cards.
- Smart card operating systems (basic file OS, MULTOS, modern Java card OS).
- Attacks (e.g. logical, physical, side channel (SPA,DPA), fault injection) and countermeasures.
- RFID tags: hardware and communication, RFID operating systems (MiFare), attacks (logical, spoofing)
- Defensive programming.
- Trusted Platform Modules (TPM) and Trusted Computing (TC)
- Misceleneous topics in OS security (e.g. hypervisors (Nova/L4Sec), MAC, AC complexities of real OSs.
- Case studies

(e) **Assessment Method:**
Assessment will be by tests and practical assignments (40%) and final examination (60%)

(f) **Reading List:**

- Ross Anderson (2001) Security Engineering: A guide building dependable distributed systems. Willey

5.2.12 MCS 7223: Combinatorial Optimization

(a) **Description:**
The course covers ways of optimizing operational systems using mathematical optimization approaches. It covers the solutions to the general set up as well as generating specific solutions to their instances.

(b) **Aims:**
The aims of the course are to equip students with techniques of optimizing real life problems using mathematical analysis and computer simulations.

(c) **Teaching and Learning Patterns:**
Teaching will be in form of lectures, class presentations and laboratory (simulated) demonstrations

(d) **Indicative Content:**

- Linear Programming extensions: Revised simplex method, Post optimality analysis.
- The transportation problem.
- Graph theory and Network analysis: Spanning trees, minimal cost networks, shortest route problem, critical path method.
- Maximal flow problems (MFP).
- Assignment (matching) problems.
- Project activity scheduling.
- Inventory theory.

(e) **Assessment Method:** Assessment will be in form of tests and practical assignments (40%) and final written examination (60%)

(f) **Reading List:**

- Combinatorial Optimization: Algorithms and Complexity by Christos H. Papadimitriou, Kenneth Steiglitz, Dover Publications, 1988

5.2.13 MCS 7224: Computer Vision

(a) **Description**
The course explores the design of computer algorithms and hardware that models the structure and properties of visual data. It explores techniques of analysis, reconstruction, digitalizing and water marking images.

(b) **Aims:**
The aim of the course is to give students an in depth understanding of the analysis and processing of digital images.

(c) **Teaching and Learning Patterns:**
Teaching will be in terms of class lectures, reading assignments and class presentations

(d) **Indicative Content:**

- Definitions, Image Types, Discretization
- Degradations in Digital Images
- Image Transformations I: Continuous Fourier Transform
- Color Perception and Color Spaces
- Image Enhancement
- Feature Extraction
- Texture Analysis
- Segmentation
- Image Sequence Analysis
- Image Reconstruction
- Object Recognition

(e) **Assessment Method:**
Assessment will be in terms of assignments and tests (40%) and final examination (60%)

(f) **Reading List:**

- R. C. Gonzalez, R. E. Woods: Digital Image Processing. Addison-Wesley, Second Edition, 2002.
- K. R. Castleman: Digital Image Processing. Prentice Hall, Englewood Cliffs, 1996.

- R. Jain, R. Kasturi, B. G. Schunck: Machine Vision. McGraw-Hill, New York, 1995.
- R. Klette, K. Schlns, A. Koschan: Computer Vision: Three-Dimensional Data from Images. Springer, Singapore, 1998.
- E. Trucco, A. Verri: Introductory Techniques for 3-D Computer Vision. Prentice Hill, Upper Saddle River, 1998.

5.2.14 MCS 7225: Image Processing

(a) **Description:**
The course explores the design of computer algorithms and hardware that models the structure and properties of visual data. It explores techniques of analysis, reconstruction, digitalizing and water marking images.

(b) **Aims:**
The aim of the course is to give students an in depth understanding of the analysis and processing of digital images.

(c) **Teaching and Learning Patterns:**
Teaching will be in terms of class lectures, reading assignents and class presentations

(d) **Indicative Content:**

- Definitions, Image Types, Discretization
- Degradations in Digital Images
- Image Transformations I: Continuous Fourier Transform
- Color Perception and Color Spaces
- Image Enhancement
- Feature Extraction
- Texture Analysis
- Segmentation
- Image Sequence Analysis
- Image Reconstruction
- Object Recognition

(e) **Assessment Method:**
Assessment will be in terms of assignments and tests (40%) and final examination (60%)

(f) **Reading List:**

- R. C. Gonzalez, R. E. Woods: Digital Image Processing. Addison-Wesley, Second Edition, 2002.
- K. R. Castleman: Digital Image Processing. Prentice Hall, Englewood Cliffs, 1996.
- R. Jain, R. Kasturi, B. G. Schunck: Machine Vision. McGraw-Hill, New York, 1995.
- R. Klette, K. Schlns, A. Koschan: Computer Vision: Three-Dimensional Data from Images. Springer, Singapore, 1998.
- E. Trucco, A. Verri: Introductory Techniques for 3-D Computer Vision. Prentice Hill, Upper Saddle River, 1998.

5.2.15 MCS 7226: Seminar Series

(a) **Description:**
The course helps students to strengthen their ability to do guided research, make a report on technical issues and present these issues in a scientific set up. While lecturers will give the students guidelines on the topics to research on, they will not formally teach them in class. However, what is expected out of the students will be explicitly given to them and examined.

(b) **Aims:**
The aims of the course are:

- To develop the students' ability to search for and internalize scientific academic material
- To develop the student's skills in technical writing
- To develop the student's presentation skills.

(c) **Learning outcomes:**
At the end of this course unit students should be able to:

- Read and internalize scientific academic material in his/her area of study
- Adequately and competently report academic findings in technical documents (reports, articles, etc)
- Prepare good presentations for dissemination of scientific findings
- Competently present scientific findings

(c) **Teaching and Learning Patterns:**
Students will be given broad areas of study together with research questions to address by the beginning of the second semester. Each student will be given a senior staff from whom they can get advice and guidance whenever necessary. The student will then be required to address one research problem and make a write up on it. The student will then be required to present his work to the staff and his/her peers. As part of the course, the student will also be obliged to attend all (weekly) research talks in the faculty (for the entire second semester).

(d) **Indicative Content:**
The content is both in terms of skill and technical content.

- Technical content: This depends on the problem addressed. The student is expected to show understanding and comprehension of the subject matter.
- Skill content: A student is expected to show ability to comprehend scientific literature, correctly make a technical report and competently prepare and make an academic presentation.

(e) **Assessment Method:**
Assessment will be made up of 4 parts:

- Attendance of weekly research talks (Semester 2) 10%
- Report write up 50%
- Presentation 20%
- Knowledge of subject matter 20%

(f) **Reading List:**
The textbooks and articles will depend on the problem being addressed.

5.3 Year I Recess Term

5.3.1 PGD 6307 Postgraduate Diploma Project in Computer Science

(a) **Description:** The Postgraduate Diploma project helps the student put into practice what has been covered in his/her area of specialisation by implementing a non trivial project. The project should address a problem of practical importance in the computerised world.

(b) **Aims:** The course aim of the course is to give a student skills in intergrating the knowledge got from different areas into something of practical usefulness.

(c) **Teaching and Learning Patterns:**
The student will work on the project under the supervison of member of staff.

(d) **Content:**
Content can varry depending on what problem the student is addressing

6 Resources and Infrastructure

The Department of Computer Science and the Faculty of Computing and Information Technology have the enough resources and infrastructure to sufficiently run the programme. Details of the resources are in Appendix C

7 Quality Assurance

Several activities will be carried out as quality assurance measures so as to

(a) Measure the general extent to which the required skills have been achieved

(b) Ascertain the Implementation of the methodological changes proposed

(c) Create a feedback benchmark for possible future revisions in the curriculum

The following activities will be carried out in the process of monitoring and assuring quality in the proposed program.

7.1 Feedback from students enrolled

In the current set up, each class has 4 student representatives (2 day, 2 evening). These representatives are in constant contact with the Head of Department in case there are any quality related matters in a particular class. This set up is to be maintained.

At the end of the semester, samples of students are given questionnaires to respond to several quality related matters like staff punctuality, delivery mode, course content and the general perceived usefulness of the course unit. The Faculty of Computing and Information Technology is the process of

creating a computerized system that will capture and analyze the data. With the computerized system:

(i) Every student will be required to assess every lecturer teaching him/her, the sample space will therefore be increased

(ii) No time will be required in the analysis of the results. Staff and faculty management will be able to get the feedback instantly

(iii) Data will be easily archived and therefore the trend of staff performance in specific areas will be easy to visualize

7.2 Class meetings

The faculty management makes at least 1 meeting with every class every semester. In this meeting, general quality issues are addressed. Students are also given a chance to raise any questions that are answered and/or addressed by the department management. This set up will also continue

7.3 Use of ICT in availing lecture materials

Currently, Makerere University has the e-learning tool on its Intra net. Initially it was Blackboard but now its Moodle. Students in the Department of Computer Science have adequate access to computers. This creates a good environment for e-learning blended teaching. All courses in the new curriculum will be taught in a blended way. All course materials will be put on the e-learning platform. Staff will, as much as possible, make use of e-learning facilities like discussion forum, blogs and drop boxes. This will increase student activity/participation and reduce staff effort (e.g. staff will not need to dictate notes). This, in turn, will increase the material covered and taken in by the students.

7.4 Peer review

All members of staff will enroll (as students) to all classes taught in the department. They will therefore be able to view contents of courses taught by their peers. Staff will be free to advise fellow staff on the content, depth and presentation of materials. Consequently, for every course, students will access the best possible material in the view of all staff in the department not the course instructor

7.5 External examiners' reports

Like it is everywhere in Makerere University, student results are reviewed every semester by a senior external academician. This is to bring a 'foreign view' of the quality of the program. External examiners write reports on their view of the curriculum/examinations. Some recommendations can be implemented immediately while others have to be implemented in a longer term. The department will make the maximum possible use of external examiners' reports as a means of assuring quality in the program.

7.6 Tracer studies

The Faculty of Computing and Information Technology is devising ways of keeping in contact with its alumni together with their employers. This is with a view of making a tracer study of its graduates. The Department of Computer Science will use outputs of the tracer studies to gauge the quality of the program and whenever necessary, improve it.

Appendix A: Staff

No	Name (Highest Qualification)	Rank	Dept
1	Venanius Baryamureeba (Ph. D)	Professor	CS/Dean
2	Irina Ya Zlotnikova (Ph. D)	Professor	IT
3	Idris Rai (Ph. D)	Ass. Professor	NW
4	Patrick J Ogao (Ph. D)	Ass. Professor	IS
5	Jose G. Quenum (Ph. D)	Senior Lecturer	CS
6	Jude T Lubega (Ph. D)	Lecturer	IT
7	Martin Bagaya (Ph. D)	Lecturer	IS
8	Josephine Nabukenya(Ph. D)	Lecturer	IT
9	Agnes R Ssemwanga (Ph. D)	Lecturer	IS
10	Benjamin Kanagwa (Ph. D)	Lecturer	NW
11	John Quinn (Ph. D)	Lecturer	CS
12	John Ngubiri (Ph. D)	Lecturer	CS