Gulu University

Faculty of Science

Department of Computer Science

P.O. Box 166 Gulu

Programme Curriculum and Regulations

For

Bachelor of Science in Computer Science Degree Programme

Week Day Programme

March, 2017

TABLE OF CONTENTS

1.0 INTRODUCTION

1.1 Gulu University

Gulu University is a Public University established in October 2002 by Statutory Instrument No. 16 of 2003. Gulu University, through its programmes, is set to make significant contributions to peace, stability, reconstruction and sustainable development.

The Vision of Gulu University is to be a leading academic Institution for the promotion of community transformation and industrialisation for sustainable development. The Mission being to expand access to higher education, provide quality professional training for delivery of appropriate social services and conduct research geared towards community transformation and conservation of biodiversity. The Vision, Mission and Goals of Gulu University are set to serve as a launch-pad for equitable and sustainable development.

The core values of Gulu University include: professionalism; integrity, effectiveness and efficiency; accountability and transparency; teamwork; gender responsiveness; concern for the elderly and people with disabilities.

The Faculty of Science was established in 2008, following the split of the former Faculty of Science Education into Faculty of Science and Faculty of Education and Humanities. The Vision and Mission of the Faculty of Science is in tandem with that of Gulu University; to play a leading role in post-war reconstruction and rehabilitation of the region through the provision of human resources in the areas of basic and applied sciences, technology, research and other services. It is also in conformity with the strategic plan of Gulu University aimed at producing high level human resource that can effectively participate in solving the social and economic problems of the country.

The Department of Computer Science is the largest of the six departments under Faculty of Science. It was established 2003 with one programme (Diploma in Computer Science). Bachelor of Science in Computer Science programme was introduced in 2005 and subsequently in 2006, Bachelors in ICT (BICT) was introduced. To-date the Department stills runs the original three programmes (Diploma, BSc. and BICT) hence the need to diversify into new programmes as demanded in the mainstream ICT industry as well as other areas where ICTs are used.

2.0 Programme Name and Corresponding Award

The name of the programme is Bachelor of Science in Computer Science (GCS) leading to an award of Bachelor of Science in Computer Science (BSc CS) of Gulu University.

3.0 Programme Duration

The duration of BSc CS programme is three (3) academic years which corresponds to six (6) semesters.

4.0 Programme Description

The programme focuses on the concepts and techniques used in the design and development of advanced software systems. Students in this program explore the conceptual underpinnings of Computer Science -- its fundamental algorithms, programming languages, operating systems, and software engineering techniques.
The programme maintains a good balance between essential theory and practice. In the theory, modules will develop students' understanding of the cognitive principles involved in user interaction with a wide range of computer systems and hardware devices.

5.0 Programme Rationale

Information and Communication Technologies (ICTs) have become a key factor in global development. For the case of developing countries, ICTs are believed to be an enabler for sustainable development (Brown & Grant, 2010). The BSc CS programme is intended to create Computer Science champions who combine the necessary technical and contextual competencies to deliver sustainable CS projects. The students will acquire long-term capabilities in development and informatics to use digital technologies to transform socio-economic development. As such, the programme will prepare students to work in highly dynamic environment of CS, including government sector, private sector, international organisations, NGOs, and global business.

5.1. Justification for Review

The IT industry is the most dynamic among the many fields of study. Changes in technology like use of Mobile phones, social computing, etc. have led to the need to revise the curriculum in order to include these emerging trends. For example, in the old curriculum a course unit like Mobile Computing focussed on old technologies like J2ME or MIDP which are now obsolete and therefore need to be upgraded.

6.0 Programme Goals, Objectives and Learning Outcomes

The goal of the BSc. CS programme is to produce competent individuals who are able to initiate and implement sustainable change with rigor in developing regions and especially in Africa, using ICTs. The objectives of this programme are:

i) To build human resource capacity in CS discipline who are able to develop Computer Systems that meet industrial, organization and society needs;
ii) To generate a pool of highly qualified candidates able to pursue research careers in MSc in CS;
iii) To produce professionals with theoretical and practical skills in CS in order to address the increasing demand of research and development;
iv) To promote social economic development by directing students' projects on topics of direct profession concern to them.
v) To prepare students for life-long learning.

6.1 Learning Outcomes

Upon successful completion of the Degree of Bachelor Science in Computer Science at Gulu University, a student will have demonstrated:

i) An ability to apply knowledge of computing and mathematics appropriate to the discipline.

ii) An ability to think critically and apply the scientific method.
iii) An ability to analyze a problem and craft an appropriate algorithmic solution.
iv) An ability to design, implement and evaluate an appropriate and secure computer-based system, process, component, or program to satisfy required specifications.

7.0 Admission Requirements

Admission into the first year is through any of the three avenues, the Direct Entry Scheme, the Mature Age Scheme and the Diploma Holders Scheme.

7.0.1 Direct Entry Scheme

Candidates seeking admission through Direct Entry Scheme must have obtained:

1. At least a principle pass in Mathematics and a principle pass at the same sitting in UACE in any of the following subjects: Economics, Entrepreneur, Geography, Physics, Chemistry, Biology, Agriculture, Technical Drawing and Food & Nutrition.

2. A minimum weighted points set by the Admissions Board of Gulu University.

For purposes of computing weighted points, the advanced level subjects shall be grouped and weighted as shown in Table 1.

Table 1: Weighting scheme for advanced level subjects.

Groups	Weights	Subjects
Essentials	3.0	Any two best done Essential subjects above.
Relevant	2.0	Any other best done Essential subjects.
Desirable	1.0	General Paper Sub-Mathematics.
Others.	0.5	All others

7.0.2 Mature Age Scheme

Admission may also be via the Mature Age Entry Scheme, after the passing of two special Mature Age Entry Examinations of Gulu University, one in aptitude and the other in specialized knowledge.

7.0.3 Diploma Holders Scheme

Holders of the Uganda National Examinations Board Ordinary Technical Diploma or its equivalent can be admitted to the programme. Applicants should have obtained:

1. At least 5 credit passes inclusive of Mathematics got at the same sitting of Uganda Certificate of Education or its equivalent.

2. At least 1 principle pass and 2 subsidiary passes from the same sitting of the Uganda Advanced Certificate of Education (UACE) or its equivalent.

3. A Credit Class diploma in Computer Science or Information Technology from a recognized Institution.

7.1 Admission to Other Years

Admission other than to the first year of the programme shall require a special resolution of the Faculty Board and permission of the Senate. The Department will work out all appropriate Credit transfers, which shall not exceed 30% of the minimum degree Credit Units.

Persons holding Diploma in Computer Science of Gulu University with at least a second class (lower division) can be admitted to 2nd year of Bachelor of Science in Computer Science of Gulu University. On the recommendation of the Faculty Board a student may be exempted from courses of similar or same content done at Diploma. The scores at the Diploma for the concerned courses shall be considered.

8.0 Available and Proposed Human Resource

The programme will engage the existing human resources at Gulu University and where necessary, external academics will be recruited. The following table gives the details of available staff who will be involved in running the courses.

S/N	Name	Qualification and Awarding Institution	Area of Specialization	Status
1	Benedict Oyo	BSc. Educ (Mak) MSc. Computer Science (Mak), PhD Information Systems (Mak)	E-learning, ICT4D, System Dynamics, Information Management, Design Science	Full-time
2	Boogere James	BSc. Educ (MUST) MSc. Computer Science (Mak)	E-learning	Full-time
3	Tabo Olok Geoffrey	BSc. Education (Mak) MSc. Computer Science (Mak) PhD –Student	Information Retrieval, Problem base E-learning	Full-time
4	Olango Proscovia	BLIS (Mak) Masters in Information Science (ITC, The Netherlands) PhD – Finalist	Information Retrieval and Dissemination, Human Interaction Computing	Full-time
5	Oketcho Moses Machulu	BIT (India) MA Business Computing	Data warehousing, Databases and Ethics in IT	Full-Time

		(India)		
6	Okot David Pakono	BSc. Computer Science (Gulu), MSc. Wireless Networking (UK)	Wireless Security, Networking	Full Time
7	Okot Patrick	BSc. Computer Science and Mathematics (Mak) MSc. Information System (Mak)	Databases, Embedded Systems, GIS & Remote Sensing, Object Oriented Programming	Full-Time
8	Ashaba Anthony Arthur	BSc. Computer Science (Mak) MSc Data Communication and Software Engineering Student (Mak) – Finalist	Computer Security, Networks, and Web Development, Programming	Full-Time
9	Guma Patterson Isdore	BSc Computer Science (Gulu) MSc. Information Systems (Mak) – Finalist	Information System, System Dynamics and Information Management	Full-Time
10	Abandu Jackson	BSc Computer Science (Gulu) MSc. Information Technology (Mak)	E-health Computing, Information Technology, Artificial Intelligence	Full-Time
11	Akello Carmela	BSc Computer Science (Gulu) MSc Information Systems (Mak) – Finalist	E-commerce, Information System Management	Full-Time
12	Kaye Milton	B.ICT (Gulu) MSc Data Communication and Software Engineering Student (Mak) – Finalist	Software Engineering, Game Development	Full-Time
13	Aguma Boniface	Bsc. Computer Science (Gulu) MSc. Data Communication and Software Engineering (Mak) – Finalist	Computer Networking, Game Development, Wireless and Mobile Communication	Full Time
14	Akello Kalumera Christine	BIT (Gulu) MIT (MUK) – Finalist	Speech Recognition and Animation Development	Full Time
15	Apio Sarah Gladys	BIT (UCU) PGD Project Planning & Mgt (UMI)	Databases, Information Systems, Project Planning & Mgt	Full Time

9.0 Infrastructure Facilities

a) Lecture Rooms/ Computer Laboratory

The Department has four Computer laboratories used for both lectures and practicals. The first Computer laboratory (Lab II), measuring 18 m by 9 m, has 40 thin client Computers. The second lab (Lab III) measures same as the first lab and has 40 desktop computers. The third lab (Cisco Lab) measures 20 m by 10 m and has eighty (80) thin client Computers. The fourth lab (GIS Lab) has 15 desktop Computers and measures 10 m by 5 m.

b) Library

The main Gulu University library will be accessible in this programme. In addition, online books and journals will be used.
Programme has 15 desktop Computers.

10.0 Programme Regulation

10.1 Course Module Assessments

The general assessment of course modules will be based on of 100 total marks with proportions as follows:-

a) Continuous coursework – 30 marks;
b) Examination – 70 marks.

However, some courses have varying assessment distributions that are given in the detailed course descriptions. A minimum of two course assignments/tests shall be required per course.

10.2 Grading System

a) **Course Unit Grading**

Each Course will be graded out of a maximum of 100 marks and assigned an appropriate letter grade and a grade point as follows:

Marks	Letter Grade	Grade Point	Interpretation
80-100	A	5.0	Excellent
75-79	B+	4.5	Very Good
70-74	B	4.0	Good
65-69	C+	3.5	Fair
60-64	C	3.0	Pass
55-59	D	2.5	Fair Pass
50-54	E	2.0	Compensatory Pass
0-49	F	0.0	Fail

b) The following additional letters will be used, where appropriate:

W:	Withdraw from Course;
I:	Incomplete;
P:	Pass;
F:	Failure.

c) **Progression**

Progression shall be regarded as normal, probationary or discontinuation as per the standard Gulu University Senate guidelines:

10.3 Pass Mark

A minimum pass grade for each course shall be 2.0 grade points.

10.4 Weighting System

The weighting unit is based on a Credit Unit (CU). A Credit Unit is one contact hour per week per semester or a series of nineteen (19) contact hours per semester. A contact hour is equal to (i) one lecture hour, or (ii) two practical hours, or (iii) two tutorial hours.

10.5 Calculation of Cumulative Grade Point Average (CGPA)

The CGPA shall be calculated as follows:

$$\text{CGPA} = \frac{\sum_{i=1}^{n} GP_i * CU_i}{\sum_{i=1}^{n} CU_i}$$

Where GP_i is the Grade Point score of a particular course i; CU_i is the number of Credit Units of course i; and n is the number of courses so far done.

10.6 Minimum Graduation Load

To qualify for the award of the Bachelor of Science in CS, a candidate is required to obtain a minimum of 75% class attendance, a CGPA of at least 2.0 from the required 131 credit units of taught courses.

10.7 Core Course

A core course is a course which is essential to a programme and gives the programme its unique features. It is offered by all the students who have registered for the programme. A core course has to be passed by every student who takes it.

10.8 Pre-Requisite Course

A pre-requisite course is a condition which must be satisfied prior to enrolling for the course in question. A pre-requisite course, therefore, is a course offered in preparation for a higher level course in the same area. A student shall not be allowed to take a higher level course unless he/she passes a pre-requisite course.

10.9 Elective Course

An elective course is a course offered in order to broaden a programme or to allow for specialization. An elective course is selected from given groups of courses of convenience of the student.

10.10 Audited Course

An audited course is a course offered by a student and whose grade point is not entered in the calculation of a GPA or CGPA. However, the GP for an audited course will be reflected in the academic transcript of the student.

10.11 Course Assessment

The course assessment shall be done by progressive assessments (like tests, assignments, group work) during the semester and final examination. The final examination may be purely written, purely practical or having a written and practical component.

(a) Each course shall be assessed on the basis of 100 total marks with proportions as follows:

- Course Work 30%
- Written Examination 70%

(b) Course work shall consist of laboratory work and progressive assessment (assignments/tests) each component assessed at 20%.

(c) For a course without laboratory work, progressive assessment shall carry 30%.

(d) A minimum of two coursework assignments and/or tests shall be required per Course.

10.12 Semester Course Load

10.12.1 Normal Semester Load

The minimum number of Credit Units per Semester shall be fifteen (15). The maximum number of Credit Units per Semester shall be twenty two (22).

10.12.2 Maximum Semester Load

The maximum number of Credit Units per Semester shall be twenty six (26) to cater for students who have courses to retake or those who are able to complete the requirements for their respective Academic Awards in less than the stipulated minimum duration.

10.13 Board of Examiners

(a) There shall be a Board of Examiners, composed of external and internal examiners appointed by Senate on the recommendation of the Faculty Board.

(b) The Board of Examiners shall receive, consider and recommend to the Faculty Board the examination results of each candidate.

(c) The Faculty Board shall receiver, consider and recommend the results of examinations to the Senate for consideration and approval.

(d) In an emergency, the Dean may act on behalf of the Faculty Board or the Board of Examiners, but must report the action taken to the next Meeting of these Boards. In so doing the Dean shall, however, act in consultation with the relevant head of Department.

10.14 Progression

Progression of a student shall be classified as Normal, Probationary or Discontinuation**.**

10.14.1 Normal Progress

Normal Progress shall occur when a student has passed all the specified Courses. This occurs when a student passes each course taken with a minimum grade point (GP) of 2.0.

10.14.2 Probationary Progress

This is a warning stage and it will occur when:

(a) A student fails the Core or Compulsory Course.

(b) A student obtains the Cumulative Grade Point Average (CGPA) of less than two (2) at the end of any semester.

(c) When the Grade Point Average of a student goes up in the following semester after the student has retaken and passed the failed Courses, then the probation shall be removed.

10.14.3 Discontinuation

(a) When a student accumulates three consecutive probations based on CGPA he/she shall be discontinued.

(b) A student who has failed to obtain at least the Pass Mark (50%) during the Third Assessment in the same Course or Courses he/she had retaken shall be discontinued from his/her studies at the University.

(c) A student who has overstayed in an Academic Programme by more than Two (2) Years shall be discontinued from his/her studies at the University.

10.15 Re-Taking a Course

(a) A student shall retake a Course when next offered again in order to obtain at least the Pass Mark (50%) if he/she had failed during the First Assessment in the Course or Courses.

(b) A student who has failed to obtain at least the Pass Mark (50%) during the Second Assessment in the same Course he/she has retaken shall receive a warning.

(c) A student may retake a Course when next offered again in order to improve his/her Pass Grade(s) got at the first Assessment in the Course were low.

(d) While retaking a Course or Courses, a student shall:

i. Attend all the prescribed lectures/Tutorials/Practicals/Fieldwork in the Course.

ii. Satisfy all the requirements for the Coursework Component in the Course.

iii. Sit for the University Examinations in the Course.

(e) When a student accumulates retakes such that his/her normal load for a semester plus the retakes exceeds the maximum semester load (i.e., 28 CU), he/she will have to Stay Put and complete the retake first. It should be noted that retakes are mandatory.

(f) A final year student whose final Examination Results have already been classified by the relevant College or Faculty or School or Institute Board and has qualified for the Award of a Degree or Diploma or Certificate, shall not be permitted to retake any Course.

(g) When a student has retaken a course, the better of the two Grades he/she obtained in that Course shall be used in the computation of his/her Cumulative Grade Point Average (CGPA).

(h) Whenever a Course has been retaken, the Academic Transcript shall indicate so accordingly.

(i) A student who does not wish to retake a failed Elective Course shall be allowed to take a substitute Elective.

10.16 Conceded Pass

10.16.1 Definition of conceded pass

A conceded pass is a pass granted for a course in which a final year candidate is within five marks of a pass mark in the course assessment. The pass is conceded on the basis that the student’s overall performance in other courses for the program has been sufficiently strong to counter the deficient percentage in that particular course.

10.16.2 Circumstances Potentially Warranting a Conceded Pass

The personal circumstances of a student must be taken into account, the student’s performance in the course could have been adversely affected by his or her personal circumstances. The circumstances for approval of a conceded pass may include but not limited to:

Student illness or medical condition

Family issues (family injury or illness, bereavement, etc)

Commitment to participate in national sport or other activities that warrant favorable consideration.

Commitment to assist with service activities.

Unavoidable and unexpected work commitments (e.g. relocation).

Awarding conceded passes does not compromise there equipment's for accreditation of that programme by a professional body.

10.16.3 Responsibility and Procedure

The conceded passes are granted at the discretion of the faculty/institute body of governors. Students are not automatically entitled to the conceded passes and may not request them.

The board of examiners shall during the time of consideration of examination results, identify and grant students the legibility for conceded passes. A student will then be formally informed that he/she has been offered a conceded pass.

10.16.4 Eligibility for Conceded Pass

A conceded pass shall be granted under the following conditions:

A candidate shall be eligible for conceded pass if the final mark in the course is in the range of 45-49%, CGPA of a student will be at least 2.0.

A conceded pass may only be awarded if a student has attempted a paper at least three times. The better of the grades and will be used for awarding a conceded pass.

A conceded pass shall be discretionary and examination boards shall take into account the following;

The result a student has scored, each time he/she has attempted a paper.

A student's overall academic record.

Comments from his/her lecturers, e.g. on his/her class attendance and participation

Whether the course is required for professional accreditation or it is necessary for a student to demonstrate professional or clinical competence as part of its assessment requirements.

A conceded pass shall be granted to a whole course not a particular piece of assessment.

Candidates granted conceded pass shall earn a credit on the basis of conceded pass "CP"

Only candidates in their final year of study shall be eligible for conceded pass.

The number of conceded pass will only be restricted to one course.

In case a candidate does not qualify for conceded pass as stipulated above, the existing provision in the semester regulation will guide as the case may be.

10.16.5 Recording a conceded pass in the academic transcript

A granted conceded pass will be recorded on the student's academic transcript by indicating a true percentage/grade achieved and "CP" as a grading code.

10.17 Absence from Examination

(a) If the Faculty Board found out that a student has no justifiable reason for having been absent from a Particular examination, such a student shall receive a fail (F) Grade for the Course(s) he/she had not sat the examination in. The Course(s) in which the Fail (F) Grade was/were awarded shall also account in the calculation of the CGPA.

(b) If the Faculty Board is satisfied that a student was absent from a final examination due to justifiable reason(s) such as sickness or loss of a parent/ guardian, and then a Course Grade of ABS shall be assigned to that Course(s). The student shall be permitted to retake the final examination when the Course would be next offered or at the next examination season, if the Lecturer concerned can make the appropriate arrangements for the examination.

(c) Certificate of Due Performance

A Certificate of Due Performance shall be awarded by the department to a student who has satisfied 75% of his/her programme load within a semester. A student who does not have coursework marks shall be denied Certificate of Due Performance. Only students who attain Certificates of Due Performance shall be allowed to sit the University Examinations.

10.18 Withdrawal

A student can apply to the Faculty Board for permission to withdraw from studies at any time of the semester. A student will be allowed only a maximum of two withdrawals in an Academic Programme and each withdrawal shall be a maximum of one academic year only.

10.19 Approval of Examination Results

Approval of all examination results will be by the Faculty Board, but the results shall not be regarded as final until they are confirmed by Senate on submission of Appropriate Pass Lists to Senate.

10.20 Publication of Examination Results

The Faculty shall publish Provisional Examination Results of candidates in every examination soon after the meeting of the Faculty Examinations Committee. The Examination Results shall be arranged and published in a manner as prescribed by the Senate.

10.21 Appeals

Any student or candidate aggrieved by a decision of the Faculty Board may appeal to the Senate Examinations Committee for reversal or moderation of the decision of the Board.

10.22 Change of Course

A student may be permitted to change Elective Course(s) in an Academic Programme in order to substitute the Elective Course(s) failed. The substitute Elective Course(s) should be within the specified Elective Course(s) for that Academic Programme.

10.23 Change of Academic Programme

(a) A student may be permitted to change from one Academic Programme to another on condition that:

i. He/She had satisfied the admission requirements for the Academic Programme applied for.

ii. He/She should not have been attending lectures/tutorials and other academic activities of the Academic Programme he/she would want to change from for more than one-half of the duration of the Programme.

iii. He/She had not been previously dismissed on disciplinary grounds from the University**.**

(b) A student permitted to change his/her Programme may be allowed to transfer the Credits from the previous Academic Programme to the new Academic Programme, provided that the Credits being transferred are relevant to the new Academic Programme.

11.0 Knowledge Areas Covered in the Curriculum

The curriculum is based on a broad knowledge areas that make up practical and resourceful information systems. The core knowledge areas are:

(i) Discrete Structures
(ii) Programming Fundamentals
(iii) Architecture and Organization
(iv) Operating Systems
(v) Net-Centric Computing
(vi) Programming Languages
(vii) Human-Computer Interaction
(viii) Graphics and Visual Computing
(ix) Intelligent Systems
(x) Information Management
(xi) Social and Professional Issues
(xii) Software Engineering
(xiii) Computational Science and Numerical Methods

12.0 Programme Structure

The following section gives the details of courses corresponding with these knowledge areas.

Year 1, Semester 1 Course Structure

CODE	COURSE NAME	LH	PH	TH	CH	CU
GCS 1101	Basic Computer Skills	30	60	-	60	4
GCS 1102	Discrete Mathematics	30	-	30	45	3
GCS 1103	Fundamentals of Information Systems	30	-	30	45	3

GCS 1104	Fundamentals of Programming	30	60	-	60	4
COM 1101	Communication Skills	30	-	30	45	3
GIT 1102	Social Computing	30	-	30	45	3
	Total Credit Units					**20**

Year 1, Semester 2 Course Structure

CODE	**COURSE NAME**	**LH**	**PH**	**TH**	**CH**	**CU**
GCS 1201	Computer Maintenance and Repair	30	60	-	60	4
GCS 1202	Introduction to Object Oriented Programming	30	60	-	60	4
GCS 1203	Research Methodology	30	-	30	45	3
GCS 1204	Introduction to Internet Programming	30	30	-	45	3
GCS 1205	Introduction to Computer Networks	30	-	30	45	3
GBC 1201	Entrepreneurship Development	30	-	30	45	3
	Total Credit Units					**20**

Year 1 Recess Semester

CODE	**COURSE NAME**	**LH**	**PH**	**TH**	**CH**	**CU**
GCS 1301	Embedded Systems Development	-	120	-	60	4
GCS 1302	Cisco Certified Entry Networking Technician (CCENT)(Audited Course)	45	60	-	75	5
	Recess Total					**09**

Year 2, Semester 1 Course Structure

CODE	**COURSE NAME**	**LH**	**PH**	**TH**	**CH**	**CU**
GCS 2101	Interactive Computer Graphics	30	30	-	45	3
GCS 2102	Data Structures, Algorithms and Analysis	30	60	-	60	4
GCS 2103	Systems Modelling	30	30	-	45	3
GCS 2104	Computer Organization and Architecture	30	30	-	45	3
GCS 2105	Organizational Behaviour	30	30	-	45	3
GCS 2106	Databases	30	60	-	60	4
	Total Credit Units					**20**

Year 2, Semester 2 Course Structure

CODE	**COURSE NAME**	**LH**	**PH**	**TH**	**CH**	**CU**
GCS 2201	Automata, Computability and Complexity	30	-	30	45	3
GCS 2202	Operating Systems	30	30	-	45	3
GCS 2203	Social & Professional Issues	30	-	30	45	3
GCS 2204	Software Engineering	30	60	-	60	4
GCS 2205	Artificial Intelligence	30	30	-	45	3
GCS 2206	Programming Project	-	60	30	60	4
	Total Credit Units					**20**

Year 2, Semester 2 Recess Course Structure

CODE	COURSE NAME	LH	PH	TH	CH	CU
GCS 2301	Industrial Training	-	150	-	75	5
	Recess Total					**5**

Year 3, Semester 1 Course Structure

CODE	COURSE NAME	LH	PH	TH	CH	CU
GCS 3101	Programming Language Translation	30	30	-	45	3
GCS 3102	Network Computing	45	30	-	60	4
GCS 3103	IT Project Management	30	30	-	45	3
GCS 3104	Introduction to Computer Security	30	30	-	45	3
Elective Courses (Any Two)						
GCS 3105	Modelling and Simulation	30	30	-	45	3
GCS 3106	Mobile Phone Application Development in Java	30	30	-	45	3
GIT 3105	Geographical Information Systems	30	30	-	45	3
	Total Credit Units					**19**

Year 3, Semester 2 Course Structure

CODE	COURSE NAME	LH	PH	TH	CH	CU
GCS 3201	Advanced Database Systems	30	30	-	45	3
GCS 3202	Parallel Computation	30	30	-	45	3
GCS 3203	Logic Programming	30	30	-	45	3
GCS 3204	Capstone CS Project	-	90	30	60	4
Elective Courses (Any Two)						
GCS 3205	Mobile Web Application Development	30	30	-	45	3
GCS 3206	Information Retrieval and Web Search	30	30	-	45	3
GCS 3207	Operations Research	30	30	-	45	3
GCS 3208	Emerging Trends in Computer Science	30	30	-	45	3
	Total Credit Units					**19**

Detailed Curriculum

Course Name:	**Basic Computer Skills**
Course Code:	**GCS 1101**
Year of Study:	**1**
Semester:	**1**
Contact Hours:	**60**
Credit Units:	**4**

Description:
This course follows the International Computer Driving License (ICDL) curriculum. It also introduces students to hardware and software practical skill development. In addition they will be exposed to hardware maintenance and services, computer assembly, identifying software and software installation.

Course objectives:

This course aims at:

i. Introducing students to main concepts of ICT at a general level.

ii. Introducing students to different parts of a computer.

iii. Developing students' knowledge and skills in using basic computer application (Microsoft Word, Excel, PowerPoint and Database).

iv. Enabling students to master computer assembly and repair

Course Learning Outcome

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

i. Demonstrate knowledge in main concepts of ICT.

ii. Demonstrate knowledge of different parts of computer

iii. Demonstrate competence in using computer applications including basic computer application (Microsoft Word, Excel, PowerPoint and Database).

iv. Demonstrate knowledge how computers are assembled and repaired.

Detailed Course Content

Concepts of ICT: **16 hours**

Hardware (hardware concepts, computer performance, memory and storage, input/output devices); software (concepts of software); networks (network types, data transfer); ICT in everyday life (electronic world, communication, virtual communities, health, environ Using the

Using the Computer and Managing Files: **14 hours**

Operating system (system setup, icons, windows); file management (files/folders, copy, move, delete, restore, searching); utilities (file compression, anti-virus); print management (printer option, Print).

Word Processing: **8 hours**
Using word processing application (working with documents, enhancing productivity); creating documents (entering text, selecting and editing); formatting (text, paragraphs, styles); objects (Create table, format table, work with graphical objects); mail merge (preparing documents for mail merge, mail merge outputs); printing (preparation for printing, check documents and print documents).

Spreadsheets: **8 hours**
Using Spreadsheet application (working with spreadsheets, enhancing productivity); cells (insert, select, edit, sort, copy, move, delete); managing worksheets (rows and columns, worksheets); formulas and functions (arithmetic formulas, functions); formatting (numbers/dates, contents, alignment, border effects); charts (create charts, editing charts); printing (setup, check, print).

Using Databases: **8 hours**
Understanding databases (key concepts, database organization, relationships, operation); using database application (working with databases, common tasks); tables (records, table design); retrieving information (main operations, queries); objects (forms); outputs (reports, data export, printing).

Presentation: **4 hours**
Using presentation application (working with presentations, enhancing productivity); developing a presentation (presentation views, slides, master slide); text (handling text, formatting, lists, tables); charts (using charts, organization charts); graphical objects (insert and manipulate charts, drawing); prepare outputs (Preparation, check and deliver).

Web Browsing and Communication: **2 hours**
The Internet (concepts and terms, security considerations); using the Browser (basic browsing, settings, navigation, bookmarks); using the Web (forms, searching); Web options (saving files, prepare and print); electronic communication (concepts and terms, security considerations, e-mail theory); using e-mail (send an email, receiving e-mail, enhancing productivity); e-mail management (Organize, address book).

Semester Total Hours **60 hours**

Study Material
Laptops with a modern operating system for which the student has administrator privileges, LCD projector, networking tools, printers and fast Internet.

Method of Delivery: The course will be taught by using lectures, hands-on demonstration, laboratory works, project works and assessments**.**

Method of Assessment:

Assignments, tests, projects and final examination. Their relative contributions to the final grade are: course work (assignments, tests, projects) 30%, final examination 70%, total 100%.

Reading List

1. Avgerou, C. Discourses (2010). ICT and Development. Information Technologies and International Development,
2. Charles J. (2005). A+ Training Guide Brook Fifth Edition
3. Mapping to the CompTIA A+ Objectives
4. Cheltenham's International Computer Driving License

Course Name:	**Discrete Mathematics**
Course Code:	**GCS 1102**
Year of Study:	**1**
Semester:	**1**
Contact Hours:	**45**
Credit Units:	**3**

Description:
Introduces the foundations of discrete mathematics as they apply to computer science, focusing on providing a solid theoretical foundation for further work. Topics include functions, relations, sets, simple proof techniques, Boolean algebra, propositional logic, digital logic, elementary number theory, and the fundamentals of counting.

Course objectives:

This course aims at:

i. Introducing students to operations associated with sets, functions, and relations.
ii. Discussing the formal methods of symbolic propositional and predicate logic.
iii. Defining Computation permutations and combinations of a set, and interpret the meaning in the context of the particular application.
iv. Designing a simple circuit using fundamental building blocks.

Course Learning Outcome
By the end this course the student will be able to**:**

i. Perform operations associated with sets, functions, and relations.
ii. Apply formal methods of symbolic propositional and predicate logic.
iii. Model problems in Computer Science using graphs and trees.
iv. Describe how formal tools of symbolic logic are used to model real-life situations, including those arising in computing contexts such as program correctness, database queries, and algorithms.

Topics:

Introduction to Logic and Proofs **5 hours**

Direct proofs; proof by contradiction; Mathematical induction.

Fundamental Structures **6 hours**

Functions (surjections, injections, inverses, composition); relations (reflexivity, symmetry, transitivity, equivalence relations); sets (Venn diagrams, complements, Cartesian products, power sets); pigeonhole principle; cardinality and accountability.

Boolean Algebra **6 hours**

Boolean values; standard operations on Boolean values; De Morgan's laws.

Basic Logic **8 hours**

Propositional logic; logical connectives; truth tables; mormal forms (conjunctive and disjunctive); validity.

Digital Logic **6 hours**

Logic gates, flip-flops, counters; circuit minimization.

Elementary Number Theory **7 hours**

Factorability; properties of primes; greatest common divisors and least common multiples; Euclid's algorithm; modular arithmetic; the Chinese Remainder Theorem.

Basics of Counting **7 hours**

Counting arguments (sum and product rule, Inclusion-exclusion principle, Arithmetic and geometric progressions, Fibonacci numbers); pigeonhole principle; permutations and combinations (basic definitions, Pascal's identity, the binomial theorem); Binomial coefficients.

Semester Total Hours **45 hours**

Study Materials

Computers and Projector with a modern operating system, LCD Projector and White boards

Method of Delivery:

The course will be taught by using lectures, hands-on demonstration, laboratory works, project works and assessments.

Method of Assessment:

Assignments, tests, projects and final examination. Their relative contributions to the final grade are: course work (assignments, tests, projects) 30%, final examination 70%, total 100%.

Reading Lists:

1. Kenneth, H, R, (2006). Discrete Mathematics and Its Applications. 5 ed. McGraw-Hill.
2. Kenneth H. R, (2006). Student's Solutions Guide to accompany Discrete Mathematics and Its Applications. 6 ed., McGraw-Hill.
3. Susanna S, E, (2003). Discrete Mathematics with Applications. Brooks Cole; 3rd ed.

Course Name:	**Fundamentals of Information System**
Course Code:	**GCS 1103**
Year of Study:	**1**
Semester:	**1**
Contact Hours:	**45**
Credit Units:	**3**

Description:

Today, information systems (IS) are an integral part of all business activities and careers. This course is designed to introduce students to contemporary information systems and demonstrate how these systems are used throughout global organizations. The focus of this course will be on the key components of information systems - people, software, hardware, data, and communication technologies, and how these components can be integrated and managed to create competitive advantage. Though the knowledge of how IS provides a competitive advantage students will gain an understanding of how information is used in organizations and how IT enables improvement in quality, speed, and agility. This course also provides an introduction to systems and development concepts, technology acquisition, and various types of application software that have become prevalent or are emerging in modern organizations and society.

Course objectives:

This Course aims at:

i. Defining how technology, people, and organizational components of information systems.
ii. Defining the value of information systems investments as well as learn to formulate a business case for a new information system, including estimation of both costs and benefits.
iii. Designing organizations develop and acquire information systems and technologies.
iv. Define how enterprise systems foster stronger relationships with customers and suppliers and how these systems are widely used to enforce organizational structures and processes.

Course Learning Outcome

Upon completion of this course, the students will acquire knowledge that will enable them to:

i. Evaluate the ethical concerns that information systems raise in society and the impact of information systems on crime, terrorism, and war.
ii. Apply Know how various types of information systems provide the information needed to gain business intelligence to support the decision making for the different levels and functions of the organization.
iii. Apply secure information systems resources, focusing on both human and technological safeguards.
iv. Demonstrate how organizations develop and acquire information systems and technologies.

Topics:

Information Systems Components **4 hours**

Hardware; software; data; networks; facilities; personnel; services**;** Partners**.**

Information Systems in Organizations 5 hours

Characteristics of IS professionals; IS career paths; cost/value information; quality of information; competitive advantage of information; IS and organizational strategy; value chains and networks.

Globalization 5 hours

What is globalization? Technology enabled change; digital divide; global information systems strategies.

Value Information Systems 8 hours

How information systems enable organizational processes; making a business case for information systems; productivity paradox of information systems; investment evaluation (multi-criteria analysis, cost-benefit analysis); identifying and implementing innovations.

Information Systems Infrastructure 7 hours

Hardware; software; collaboration and communications technologies; data and knowledge; networks; facilities; personnel; services; partnerships.

The Internet and WWW 7 hours

E-Business (B2C, B2B); Intranets, Internet, Extranets; e-government; Web 2.0 - technologies (e.g., wikis, tags, blogs, net casts, self-publishing), new forms of collaboration (social networking, virtual teams, viral marketing, crowd-sourcing).

Securing Information Systems 4 hours

Threats to information systems; technology-based safeguards; human based safeguards; information systems security planning and management.

Gaining Business Intelligence from IS 6 hours

Organizational decision making, functions, and levels (executive, managerial, and operational levels; systems to support organizational functions and decision making); information and knowledge discovery (reporting systems; online analytical processing; data, text, and web mining; business analytics); application systems (executive, managerial, and operational support systems; decision support systems; functional area information systems; collaboration technologies; intelligent systems; knowledge management systems); information visualization (visual analytics; dashboards; geographic information systems).

Enterprise Wide IS 6 hours

Enterprise resource planning; supply chain management; customer relationship management.

Developing and Acquiring IS Resources 5 hours

Systems development lifecycle; alternative development approaches; external acquisition; outsourcing; end-user development.

IS Ethics and Crimes 2 hours

Information privacy, accuracy, property, and accessibility; computer crime; cyber war / cyber terrorism.

Semester Total Hours 45 hours

Study Materials
Computers and Laptops with a modern operating system, Projector and White boards

Method of Delivery:
The course will be taught by using lectures, case-studies, laboratory work, class projects and assignments.

Method of Assessment:
Assignments, tests, projects and final examination. Their relative contributions to the final grade are: course work (assignments, tests, projects) 30%, final examination 70%, total 100%.

Reading Lists:

1. Avison D & Fitzgerald G (2011). Information Systems Development: Methodologies, Techniques and Tools. 4th ed., McGraw-Hill Higher Education**.**
2. Kenneth C (2011). Laudon and Jane P. Laudon. Essentials of Management Information Systems. 9th ed., Prentice Hall.
3. O'Brien J A. and George M. M (2008). Introduction to Information Systems. 14th ed., McGraw-Hill Irwin

Course Name:	**Fundamentals of Programming**
Course Code:	**GCS 1104**
Year of Study:	**1**
Semester:	**1**
Contact Hours:	**60**
Credit Units:	**4**

Description:
The course introduces the fundamental concepts of procedural programming, including data types, control structures, functions, arrays, files, and the mechanics of running, testing, and debugging. The course also offers an introduction to the historical and social context of computing and an overview of computer science as a discipline.

Students should have sufficient facility with high-school mathematics to solve simple linear equations and to appreciate the use of mathematical notation and formalism.

Course objectives
This course provides students with

i. Skills to produce algorithms for solving simple problems and trace the execution of computer programs.
ii. Constructing object-oriented, structured, and functional programming methodologies.
iii. Designing the language translation phases of compiling, interpreting, linking and executing, and differentiate the error conditions associated with each phase.

Course Learning Outcome
Upon completion of this course, the students will acquire knowledge that will enable them to:

i. Construct a simple program for solving simple problems and trace the execution of computer programs.
ii. Apply the program development process to problems that structured, and functional programming methodologies.
iii. Decompose a program into subtasks and use parameter passing to exchange information between the subparts.

Topics:

Fundamental Programming Constructs: 15 hours

Syntax and semantics of a higher-level language; variables, types, expressions, and assignment; simple I/O; conditional and iterative control structures; functions and parameter passing; structured decomposition.

Algorithms and Problem-Solving: 15 hours

Problem-solving strategies; the role of algorithms in the problem solving process; implementation strategies for algorithms; debugging strategies; the concept and properties of algorithms.

Fundamental Data Structures: 10 hours

Primitive types; arrays; records; strings and string processing.

Machine Level Representation of Data: 5 hours

Bits, bytes, and words; numeric data representation and number bases; representation of character data.

Overview of Operating Systems: 5 hours

The role and purpose of operating systems; simple file management.

Software Development Methodology: 5 hours

Fundamental design concepts and principles; structured design; testing and debugging strategies; test-case design; programming environments; testing and debugging tools.

Social Context of Computing: 5 hours

History of computing and computers; evolution of ideas and machines; social impact of computers and the Internet; professionalism, codes of ethics, and responsible conduct; copyrights, intellectual property, and software piracy.

Semester Total Hours 60 hours

Study Material

Laptops with a modern operating system for which the student has administrator privileges, LCD projector, networking tools, printers and fast Internet. Computers and Laptops with a modern operating system and JAVA and JDK platform, Projector and White boards

Method of Delivery:

The course will be taught by using lectures, hands-on demonstration, laboratory works, project works and assessments.

Method of Assessment:

Assignments, tests, projects and final examination. Their relative contributions to the final grade are: course work (assignments, tests, projects) 30%, final examination 70%, total 100%.

Reading Lists:

1. Allen B. Downey (2008). How to Think Like a Computer Scientist (Java Version). Olin College, Needham, Massachusetts.
2. Allen, B. D (2009). Python for Software Design: How to Think Like a Computer Scientist. 1st ed., Cambridge University
3. Dale N, W C & Headington M (2001). Programming in C++, Second Edition. Jones and Bartlett Publishers, Inc.
4. David J. E (2006). Introduction to Programming Using Java, Fifth Edition.
5. Deitel. P. J. and Deitel H. M. (2007) How to Program. Seventh Edition. Pearson Education Inc.
6. Jennifer, C (2009). Practical Programming: An Introduction to Computer Science Using Python (Pragmatic Programmers). 1st ed., Pragmatic Bookshelf.
7. Khalid A. M, Torill. H and Rolf W. R (2007). Java Actually. A First Course in Programming. Thomson Learning.
8. Zelle, J (2003). Python Programming: An Introduction to Computer Science.

Course Name:	**Communication Skills**
Course Code:	**COM 1101**
Year of Study:	**1**
Semester:	**1**
Contact Hours:	**45**
Credit Units:	**3**

Description:

A widely heard theme among employers is that computer science professionals must be able to communicate effectively with colleagues and clients. Because of the importance of good communication skills in all computing careers, computer science students must sharpen their oral and writing skills in a variety of contexts – both inside and outside of computer science courses. This course introduces to the students principles of organization, development, and writing of technical documents; and instils in them skills of listening, speaking and interaction.

Course objectives:

The purpose of this course is to:

i. Introduce students to skills of reading,
ii. Develop listening and speaking and interaction.
iii. Create technical, writing and documentation skills.
iv. Develop public and formal presentation skills.

Course Learning Outcome

By the end of this course students:

i. Should be able to gain knowledge in skills of reading

ii. Have ability to listen, speak and interact freely
iii. Demonstrate writing and documentation skills.
iv. Enhance their public and formal presentation skills.

Detailed Course Content

Interpersonal Skills: 15 hours

Reading both individual and public; listening skills; speaking, interaction, and conversational skills; the concept team work; interoffice and intra-office communication; conduct of discussions and dynamics of meetings.

Writing and Documentation Skills: 15 hours

Note-taking; writing minutes; writing notice of meeting and agenda; preparing formal documents (resume, application letters, acceptance letters, resignation letters, memos, circulars, responses, letters of introduction etc.); development of technical and academic documents(theses, proposals, dissertations, laboratory reports, papers, articles, abstracts).

Oral Presentation Principles: 15 hours

Visual and computer-assisted presentation; analysis and design of Web presentation; choice and use of appropriate presentation tools; organizing and presenting effective talk**.**

Semester Total Hours 45 hours

Study Materials

Computers and Projector with a modern operating system and White boards

Method of Delivery:

The course will be taught by using lectures, hands-on demonstration, laboratory works, project works and assessments.

Method of Assessment:

Assignments, tests, projects and final examination. Their relative contributions to the final grade are: course work (assignments, tests, projects) 30%, final examination 70%, total 100%.

Reading Lists:

1. Perkins P, S. (2008) The Art and Science of Communication: Tools for Effective Communication in the Workplace.
2. Ros J. (2003) How to Write Proposals & Reports that Get Results. Pearson Prentice Hall.
3. Saleemi N. A. (1997). Business Communication and Report Writing Simplified.

Course Name:	**Social Computing**
Course Code:	**GIT 1103**
Year of Study:	**1**
Semester:	**1**
Contact Hours:	**45**
Credit Units:	**3**

Description:

This course introduces students to the major opportunities and challenges for creating online communities. The course attempts to examine the future of the Internet in the context of collaboration experiences that go beyond what's possible face to face. The view of people as social creatures and the context of current social platforms are also covered in this course.

Course objectives:

This course aims at:

i. Introducing students to opportunities and challenges of online communities.
ii. Supporting student to gain understanding of values of social media platforms.
iii. Enabling students to develop mastery of social computing concepts and applications areas.
iv. Imparting knowledge in security and privacy issues in social computing.

Course Learning Outcomes

Upon completing this course the students will be able to:

i. Demonstrate understanding of opportunities and challenges of online communities.
ii. Provide analysis of the values of social media platforms.
iii. Show competence in using social media applications.
iv. Demonstrate understanding of security and privacy issues in social computing.

Topics:

Introduction to social computing	**3 hours**
Social studies of ICTs	**3 hours**
Information and collaboration in organizations	**6 hours**
Current trends in social computing (social network sites, blogs, wikis and resource sharing)	**6 hours**
Semantics in social networking	**3 hours**
Crowdsourcing	**3 hours**
Computer-mediated communication and collaborative technologies	**3 hours**
Social marketing and monetization of the web	**6 hours**
Security and privacy issues related to social computing	**6 hours**
Information society	**3 hours**
Publishing digital media content	**3 hours**
Semester Total Hours	**45 hours**

Study Materials

Computers and Laptops with a modern operating system, Projector and White boards

Method of Delivery: The course will be taught by using lectures, group and social network discussions, seminars and assignments.

Method of Assessment: Assignments, tests, class presentations and final examination. Their relative contributions to the final grade are: course work
(Assignments, tests, presentations) 30%, final examination 70%, total 100%.

Reading List

1. Ozok, A.A., & Zaphiris, P. (2009).Online Communities and Social Computing. Springer: San Diego.
2. Dasgupta, S. (2009). Social Computing: Concepts, Methodologies, Tools, and Applications. IGI Global: Hershey-New York.

Course name:	**Computer Maintenance and Repair**
Course Code:	**GCS 1201**
Year of study:	**1**
Semester:	**2**
Contact Hours:	**60**
Credit Units:	**4**

Description:
The course give students advanced knowledge in repair and maintenance of computers. Students go beyond assembling into identification of faulty parts and replacements as well as software installation, upgrades and troubleshooting.

Course objectives:
The course aims at:

i). Developing and demonstrating essential workplace, communication, customer service skills and business ethics
ii). Identify preventative maintenance, safety, and environmental issues, Diagnose and troubleshoot common hardware devices and operating system problems
iii). Demonstrating installation, configuration, and upgrading of computer hardware devices and operating systems and disassembly and assembly of computers systems
iv). Distinguishing various types of motherboards and enclosure form factors, processors, memory, internal/external hard drives, expansion slots
v). Identifying basic networking concepts, services, protocols and capabilities, Network and computer security

Course Learning Outcomes:
By the end of the course, students will be able to:

i). Define information technology (IT) and describe the components of a personal computer and apply Protect herself or himself against accidents and injury, protect equipment from damage, and protect the environment from contamination. Perform a step by step assembly of a desktop computer tower. Apply good communication skills and professional behavior while working with customers.
ii). Explain the purpose of preventive maintenance and identify the elements of the troubleshooting process
iii). Explain, install, and navigate an operating system; upgrade components based on customer needs and perform preventive maintenance and advanced troubleshooting.
iv). Describe, remove, and replace select components of a laptop, printer/scanner; upgrade components based on customer needs and perform preventive maintenance and advanced troubleshooting, install a network; upgrade components based on customer needs and perform preventive maintenance and advanced troubleshooting.
v). Perform advanced installation of a desktop computer tower; select components based on customer needs and perform preventive maintenance and advanced troubleshooting.

Upgrade security components based on customer needs and perform preventive maintenance and advanced troubleshooting.

Detailed course Content:

Introduction to the Personal Computer **4 Hours**
Describe a personal computer system

Lab Procedures and Tool Use **2 Hours**
Demonstrate safe lab procedures and proper tool use

Computer Assembly **6 Hours**
Demonstrate safe lab procedures and proper tool use.

Overview of Preventive Maintenance and Troubleshooting **2 Hours**
Explain the purpose of and basic rules of preventive maintenance and the troubleshooting process.

Operating Systems **6 Hours**
Install, upgrade, navigate and troubleshoot various operating systems.

Networks **6 Hours**
Describe, create and maintain a network.

Laptops **8 Hours**
Describe how laptops are constructed, their basic configuration, maintenance, and troubleshooting.

Mobile Devices **8 Hours**
Describe operating systems, basic configuration, maintenance, security and repair for mobile devices.

Printers **4 Hours**
Demonstrate all procedures required for printers.

Security **4 Hours**
Explain importance of security, describe procedures, and perform security measures.

The IT Professional **4 Hours**
Describe the communication and related skills needed by an IT professional.

Advanced Troubleshooting **6 Hours**
Perform advanced troubleshooting procedures on computers, operating systems, laptops, printers, network and security.

Semester Total Hours **60 hours**

Study Materials:

Laptops with a modern operating system for which the student has administrator privileges, LCD projector, printers and fast Internet. Hands-on lab activities and virtual learning tools are essential elements that are integrated into the curriculum. The Virtual Laptop and Virtual Desktop are standalone tools that enable students to virtually disassemble and reassemble desktop and laptop computers. Other tools include Packet Tracer, Technician's Toolkit, and Networking Toolkit.

Method of Delivery:
Teaching will be largely by practical demonstrations supplemented with some class lectures.

Method of Assessment:
Assignments, tests, projects and final examination. Their relative contributions to the final grade are: course work (assignments, tests, projects) 30%, final examination 70%, total 100%.

Reading List

1. Charles J. B. (2005). A+ Training Guide Book Fifth Edition;
2. Cisco Networking Academy. IT Essentials 5.0;
3. Winn L. Rosch (2003). The Hardware Bible, 6^{th} Edition;
4. Cisco IT Essentials I & II

Course Name:	**Introduction to Object Oriented Programming**
Course Code:	**GCS 1202**
Year of Study:	**1**
Semester:	**2**
Contact Hours:	**60**
Credit Units:	**4**

Description:
Introduces the concepts of object-oriented programming to students with a background in the procedural paradigm. The course begins with a review of control structures and data types with emphasis on structured data types and array processing. It then moves on to introduce the object-oriented programming paradigm, focusing on the definition and use of classes along with the fundamentals of object-oriented design. Other topics include an overview of programming language principles, simple analysis of algorithms, basic searching and sorting techniques, and an introduction to software engineering issues.

Course objectives:
This course aims at:

i. Define object oriented programming solutions for reuse, using ADTs that incorporate encapsulation, data abstraction, and information hiding.
ii. Define multiple-file or multiple-module programming solutions that use class hierarchies, inheritance, and polymorphism to reuse existing design and code.
iii. Verifying program correctness through the development of sound test plans and the implementation of comprehensive test cases.
iv. Create programming solutions that use data structures and existing libraries the execution of searching and sorting algorithms..

v. Create programming solutions that interacts with database.

Course Learning Outcome

By the end students should:

i. Construct object oriented programming solutions for reuse, using ADTs that incorporate encapsulation, data abstraction, and information hiding.
ii. Construct multiple-file or multiple-module programming solutions that use class hierarchies, inheritance, and polymorphism to reuse existing design and code.
iii. Apply program correctness through the development of sound test plans and the implementation of comprehensive test cases.
iv. Analyze programming solutions that use data structures and existing libraries the execution of searching and sorting algorithms
v. Design and develop secure databases

Review of: **5 hours**

Control structures; functions; primitive data types.

Object Oriented Programming; **20 hours**

Object-oriented design; encapsulation and information hiding; separation of behavior and implementation; classes, subclasses, and inheritance; polymorphism; class hierarchies.

Fundamental Computing Algorithms: **5 hours**

Simple searching and sorting algorithms (linear and binary search, selection and insertion sort).

Fundamentals of Event-Driven Programming; **4 Hours**

Event-handling methods; event propaGCSion; exception handling.

Introduction to Computer Graphics: **6 hours**

Using a simple graphics API.

Overview of Programming Languages: **4 hours**

History of programming languages; brief survey of programming paradigms.

Virtual Machines: **2 hours**

The concept of a virtual machine; hierarchy of virtual machines; intermediate languages.

Introduction to Language Translation: **8 hours**

Comparison of interpreters and compilers; language translation phases; machine-dependent and machine-independent aspects of translation.

Introduction to Database Systems: **10 hours**

History and motivation for database systems; use of a database query language.

Software evolution: **6 hours**

Software maintenance; characteristics of maintainable software; reengineering; legacy systems; software reuse.

Semester Total Hours **60 hours**

Study Material
Laptops with a modern operating system for which the student has administrator privileges, LCD projector, networking tools, printers and fast Internet.

Method of Delivery:
The course will be taught by using lectures, group discussions, seminars, field works and assignments.

Method of Assessment:
Assignments, tests, research assignments and final examination. Their relative contributions to the final grade are: course work (assignments, tests, projects) 30%, final examination 70%, total 100%.

Reading Lists:
1. P. J. Deitel and H. M. Deitel (2007). Jave How to Program. Seventh Edition. Pearson Education Inc.,
2. Allen B. Downey (2008). How to Think Like a Computer Scientist (Java Version). Olin College, Needham, Massachusetts,.
3. David J. Eck (2006). Introduction to Programming Using Java, Fifth Edition.
4. Nell Dale, Chip Weems and Mark Headington (2001). Programming in C++, Second Edition. Jones and Bartlett Publishers, Inc.,
5. Khalid A. Mughal, Torill Hamre and Rolf W. Rasmussen (2007). Java Actually. A First Course in Programming. Thomson Learning,
6. Zelle, John (2007). Python Programming: An Introduction to Computer Science. Franklin, Beedle & Associates. ISBN: 9781887902991.
7. Allen B. Downey (2009). Python for Software Design: How to Think Like a
8. Computer Scientist. 1st ed., Cambridge University Press.
9. Jennifer Campbell (2009). Practical Programming: An Introduction to Computer Science Using Python (Pragmatic Programmers). 1st ed., Pragmatic Bookshelf.

Course Name:	**Research Methodology**
Course Code:	**GCS 1203**
Year of Study:	**1**
Semester:	**2**
Contact Hours:	**45**
Credit Units:	**3**

Description:
This course enables students to develop methods, techniques and competencies necessary for efficient and effective research in Computer Science. A particular attention will be paid to selection, assessment and review of academic and other relevant literature. The course also provides an opportunity for students to practice critical appraisal and awareness skills, critical thinking, and develop confidence and independence in researching a given or chosen topic.

Course objectives:

The objectives of this course is to:

i. Introduces the processes, purpose and goals of literature review/ survey.
ii. Define variety of Computer Science research methodologies, and choose appropriate methodology relevant to the research issue or topic.
iii. Introduce students to appropriate tools and techniques to perform an effective and efficient academic literature search, including restricting searches, identifying, locating, retrieving and storing documents.
iv. Critically assess the relevance and impact of retrieved literature, and choose appropriate documents and citation to support the construction of own arguments.

Course Learning Outcome

By the end of this course students will be able

i. Demonstrate an ability to cite literature review.
ii. Appreciate and compare a variety of Computer Science research methodologies, and choose appropriate methodology relevant to the research issue or topic
iii. Choose appropriate tools and techniques to perform an effective and efficient academic literature search, including restricting searches, identifying, locating, retrieving and storing documents.
iv. Demonstrate an understanding of the process of synthesizing the knowledge acquired from the reviewed works

Topics:

Research Purpose and Products: 5 hours

Reasons for doing research; possible products the outcomes of research; finding and choosing research topics; evaluating research purpose and products.

Research Process: 18 hours

Research strategies; data generation methods; data analysis.

Internet Research: 4 hours

The Internet and WWW; Internet research topics; the Internet and literature review; the Internet and research strategies and methods; the Internet research, the law and ethics.

Research Ethics: 4 hours

Law and research; rights people directly involved; responsibility of an ethical researcher; design and creation projects and ethics; Internet research and ethics; evaluating research ethics.

Literature Search, Retrieval and Storage: 7 hours

Strategies and methods for effective literature search, retrieval and storage.

Appraising, Arguing and Reviewing: 3 hours

Appreciation of the process and principles of critical appraisal, construction of argument, peer reviewing.

Case Studies: 4 hours

Defining case studies; planning and conducting case studies; the Internet and case studies; examples of case studies in Computer Science; evaluating case-study based research.

Semester Total Hours **45 hours**

Study Material

Laptops with a modern operating system for which the student has administrator privileges, LCD projector, networking tools, printers and fast Internet.

Method of Delivery:

The course will be taught by using lectures, group discussions, seminars, field works and assignments.

Method of Assessment:

Assignments, tests, research assignments and final examination. Their relative contributions to the final grade are: course work
(Assignments, tests, projects) 30%, final examination 70%, total 100%.

Reading Lists:

1. Briony J. Oates (2005). Researching Information Systems and Computing. 1st ed., Sage Publications Ltd.
2. Lawrence F. Locke, Stephen J. Silverman and Waneen Wyrick Spirduso (2004). Reading and Understanding Research. 2nd ed., Sage Publications.

Course Name:	**Introduction to Internet Programming**
Course Code:	**GCS 1204**
Year of Study:	**1**
Semester:	**2**
Contact Hours:	**60**
Credit Units:	**4**

Description:

IT applications are increasingly web-based. Web technology has grown to include a variety of business, academic, organizational and social applications. Diverse multi-cultural and multi-lingual user communities now depend on Web technology. This course covers the design, implementation and testing of web-based applications including related software, databases, interfaces and digital media. It also covers social, ethical and security issues arising from the Web and social software.

Course objectives:

i. Describe the structure of the World Wide Web as interconnected hypertext documents.
ii. Describe the importance of the HTTP protocol in Web applications.
iii. Create and validate HTML/XHTML documents.
iv. Demonstrate an understanding of XML syntax and show how to display such documents in Web applications.
v. Use some presentation technologies, such as Cascading Style Sheets and DHTML.

vi. Discuss and contrast data entry and validation techniques in client-side vs. server-side programming.

Course Learning Outcomes

Upon completion of the course, students will be able to:

i. Build a simple web site that organizes information effectively.
ii. Discuss the use of proprietary media and interaction technologies such as Flash, Active X, Real Media, and QuickTime.
iii. Use cascading style sheets to create style standards for a web site.
iv. Explain the importance of interfacing web sites to underlying databases.
v. Describe the methods of security for cookies.
vi. Describe the use of public key encryption to enhance security.

Topics:

HTML, XHTML and Web Site Design: 5 hours

Review of XHTML; Web pages; using images on XHTML; hyperlinks, tables, and frames; publishing your page on the Internet and search engines.

Cascading Style Sheet (CSS): 5 hours

What is CSS style? controlling text properties with style; background and border; dimensioning and positioning of elements; other design techniques using the CSS style.

Images, Animations and Multimedia: 5 hours

Using multimedia on the Web; basic multimedia applications on the Web; embedding and controlling WMP; using plug-ins for multimedia applications.

XML and XSLT: 7 hours

Introduction to XML and XSLT; transforming XML to XHTML using XSLT; manipulating XML data with XSLT; using CSS and parser on XML documents.

Client Side Scripting with JavaScript: 5 hours

Statements; comments; variables; operators; comparisons; if/else; switch; popup boxes; functions; for loop; while loop; break loops; for ... in; events; try/catch; throw; string; date; array; boolean; math; RegExp.

Server Side Scripting with PHP: 5 hours

PHP Syntax, variables, string, operators, if/else, switch, arrays, while loops, for loops, functions, forms, GET, POST; ODBC; Simple XML.

TCP/IP: 2 hours

TCP/IP overview; TCP/IP addressing; TCP/IP protocols; TCP/IP email.

Web Media: 3 hours

Multi-media overview; sound formats; video formats; playing sounds; playing videos; windows formats; QuickTime; Real Video.

Network Management: 2 hours

Overview of the issues of network management; use of passwords and access control mechanisms; domain names and name services; issues for Internet service providers; security issues and firewalls.

Compression and Decompression: 3 hours

Analog and digital representations; overview of encoding and decoding algorithms; lossless and lossy compression.

Network Security: 3 hours

Fundamentals of cryptography; secret-key algorithms; public-key algorithms; authentication protocols; digital signatures.

Software Tools and Environments: 2 hours

Web-page development tools.

Intellectual Property: 2 hours

Foundations of intellectual property; copyrights, patents, and trade secrets; issues regarding the use of intellectual property on the web.

Semester Total Hours 45 hours

Study Material

Laptops with a modern operating system for which the student has administrator privileges, LCD projector, networking tools, printers and fast Internet.

Method of Delivery:

The course will be taught by using lectures, group discussions, seminars, field works and assignments.

Method of Assessment:

Assignments, tests, research assignments and final examination. Their relative contributions to the final grade are: course work (Assignments, tests, projects) 30%, final examination 70%, total 100%.

Reading Lists:

1. P.K. Yuen and V. Lau (2003). Practical Web Technologies. Addison Wesley.
2. Chris Bates (2000). Web Programming. J. Wiley and Sons.
3. H.M. Deitel, P.J. Deitel and A.B. Goldberg (2004). Internet & World Wide Web: How to Program. Prentice Hall.
4. R. Lerdorf, K. Tatroe and P. Maclntyre (2002). Programming PHP. O’Reilly.
5. Laura Lemay and Rafe Colburn (2003). Sams Teach Yourself Publishing HTML & XHTML in 21 Days. Sams.Net.
6. Thomas A. Powell. HTML (2000). The Complete Reading List. OsborneMcGraw- Hill.
7. Jenifer Niederst (2001). Web Design in a Nutshell. O’Reilly and Associates Inc.

Course Name:	**Introduction to Computer Networks**
Course Code:	**GCS 1205**
Year of Study:	**1**
Semester:	**2**
Contact Hours:	**45**
Credit Units:	**3**

Description:

Introduces the structure, implementation, and theoretical underpinnings of computer networking and the applications that have been enabled by the state-of-the-art technology. Specifically, the course will cover Introduction to Networks: dentition, advantages, types, configurations; The OSI/ISO reference model; Transmission media: magnetic media, twisted pair, coaxial, be optics; Data encoding: straight, Manchester, deferential Manchester, satellite; Digital versus Analog transmission; Modems, modulation and their standards, codes and pulse code modulation; Integrated Services Digital Networks (ISDN); Network Access Protocols; Passive versus dynamic allocation; LAN standards:802.3 (Ethernet), 802.4 (token bus), 802.5 (token ring); Computer Network security, Active and Passive Attacks; Network layer and Network layer protocols; Transport layer and Transport layer protocols.

Course objectives:

The aims of the course are:-

i. To provide a solid basis on the theoretical and practical understand of data communication networks

ii. To introduce students to standards and guidelines in computer and data communication networks

iii. To impact knowledge and skill relevant for the design, implementation and maintenance of modern computer communication networks

Course learning outcome:

On completion of this course unit, the students will be able to:

i. Show the terminology and concepts of the OSI reference model and the TCP/IP reference model and Demonstrate knowledge of wireless networking concepts

ii. Master the concepts of protocols, network interfaces, and design/performance issues in local area networks and wide area networks;

iii. Demonstrate contemporary issues in networking technologies and; Demonstrate knowledge of network tools.

Topics:

Data Communication Concepts **8 hours**

Networks and open system standards: the OSI reference model
Network topologies and the physical layer Bus/Tree topology, ring topology, star topology
The future of data communications

Transmission Media and Transmission Technologies **7 hours**

The electrical interface Metallic media; Optical fibre media Wireless media (line-of-sight media)
Baseband and broadband transmission bandwidth (link capacity) Codes
Analog and digital signals Modulation and demodulation, modems and modem standards

Transmission impairments (distortion and noise limitations on system performance)

Data Transmission **6 hours**
Data compression Protocol Concepts - Media Access Control
Protocol basics MAC protocols (CSMA/CD and Token passing)

Data Security and Integrity **4 hours**
Error detection and correction Encryption and decryption
Viruses, worms, and hacking

Local Area Networks **6 hours**
LAN standards (IEEE standards 802 for LANs) Interconnecting LANs
LAN Hardware (server platforms, backup devices, LAN adapters, printers, etc.)
LAN system software, LAN application software LAN selection criteria

Metropolitan Area Networks (MANs) and Wide Area Networks (WANs) **6 hours**
Network routing Public data networks Circuit-switched data network Packet-switched data network; Internet protocol ISDN Electronic mail

Network Architecture **4 hours**
Layered approach & Hierarchical approach

Network Interconnections (Internetworking) **4 hours**
LAN-to-LAN connections and LAN-to-Host connections Repeaters, Bridges, Routers, and Gateways Interconnection utilities

Total Hours **45 hours**

Study Materials
Desktops/Laptops with a modern operating system, Packet Tracer, Projector and White board.

Method of Delivery:
The course will be taught by using lectures, group discussions, seminars, field works and assignments.

Method of Assessment:
Assignments, tests, research assignments and final examination. Their relative contributions to the final grade are: course work (Assignments, tests, projects) 30%, final examination 70%, total 100%.

Reading List

1. Bertsekas, Dimitri, and Robert G. (2012). Data Networks (5th Edition). Upper Saddle River, NJ: Prentice Hall.
2. Peterson & Davie (2011). Computer Networks (5th Edition). San Francisco, CA: Morgan.

Course Name:	**Entrepreneurship Development**
Course Code:	**GCS 1206**
Year of Study:	**1**
Semester:	**2**
Contact Hours:	**45**
Credit Units:	**3**

Description:
Entrepreneurship is a specialized business course designed to provide students the skills needed to effectively organize, develop, create, and manage their own business. This course is based upon the Marketing Education;
Framework which includes business, management, and entrepreneurship; communication and interpersonal skills; economics; and professional development foundations. Emphasis is placed on the functions of marketing: distribution, financing, marketing information management, pricing, product/ service management, promotion, and selling. Additional topics to be addressed are assessment of personal skills, the components of the free enterprise system and its place in our global economy, human relations and interpersonal skills, the importance of business ethics, and the role quality and service play in business. Students will develop a written business plan for a business of their choice.

Course objectives:

i. Appreciate the challenges of entrepreneurial actions in international context.
ii. Craft a draft business plan for future use in seeking venture capital and other supports.
iii. Familiarize themselves with classical elements of a business plan and develop skills in understanding the factors that venture capitalists look for in evaluating such plans.
iv. Critically analyze entrepreneurial ventures from historical and field literature to identify causal factors in success or failure of such ventures.

Course Learning Outcome

i. Describe strategies for nurturing or growing a business.
ii. Appreciate the levels and impart of risk taking to a business.
iii. Identify distinctive bases of sustainable competitive advantage that are essential to the success of an entrepreneurial firm.
iv. Identify and describe the major steps and requirements for starting a small-scale business.

Topics:

Entrepreneurship defined: 7 hours
Venture, venture capitalist, and adventure; goal setting/planning; risk taking/resource management; the role demands of business; reasons for failure (internal and externals factors).

Variety in entrepreneurship: 7 hours
Concept of “change agent” and “entrepreneurship”; a manager versus an entrepreneur; pros and cons of working for self-versus others; social ventures; profit motive versus a dream.

Characteristics of successful entrepreneurs share: 5 hours
Growth of women entrepreneurs; factors most and least important to entrepreneurs; survey – are you an entrepreneur?

Entrepreneurial skills: **6 hours**
Seven steps to a successful venture; trying a venture; finding a mentor.

Innovation: **7 hours**
The challenges of innovation and how to encourage it; student project.

Identifying and assessing opportunity: **5 hours**
Profiling entrepreneurs; market research; student project.

You – the idea machine: **3 hours**
Examining the creative process; idea generation; left and right brain thinkers; student project.

Planning your venture: **5 hours**
Goal setting – vision; financial planning – costs, break-even, statements; market strategy – target market, surveys, advertising.

Semester Total Hours **45 hours**

Study Material
Laptops with a modern operating system for which the student has administrator privileges, LCD projector, networking tools, printers and fast Internet.

Method of Delivery:
The course will be taught by using lectures, group discussions, seminars, field works and assignments.

Method of Assessment:
Assignments, tests, research assignments and final examination. Their relative contributions to the final grade are: course work (Assignments, tests, projects) 30%, final examination 70%, total 100%.

Reading Lists:

1. Peter F. Drucker (2006). Innovation and Entrepreneurship. Collins Business.
2. Bruce Barringer and Duane Ireland (2007). Entrepreneurship: Successfully Launching New Ventures. 2nd ed., Prentice-Hall.
3. Robert Hisrich, Michael Peters and Dean Shepherd (2006). Entrepreneurship. 7th ed., McGraw-Hill/Irwin.

Course Name:	**Embedded System Development**
Course Code:	**GCS 1301**
Year of Study:	**2**
Semester:	**3**
Contact Hours:	**60**
Credit Units:	**4**

Description:

This course will cover the basics of embedded system organization, system on programmable-chip technologies and real-time systems. It provides the advance knowledge required for embedded computer design and development as well as real-time operating systems. Students are introduced to software development concepts applicable to real-time and embedded systems. Particularly ARM Cortex M3 will be studied as a representative embedded processor and embedded software development is carried out for ARM Cortex CPUs. The students will be able to grasp the main principles of embedded system design and understand the concept of hardware-software co design, system on programmable chip (SoPC), real-time operating systems and scheduling techniques. Embedded system co-specification and partitioning is also introduced in the course. System C or other languages (e.g. UML, C, etc.) can be employed to present a unified view of the embedded systems. System C is introduced as a representative Co-specification language. Embedded hardware-software design and development tools (such as Altera Quartus II and SOPC builder) will be introduced.

Course objectives:

This course aims to:

i. Identify the basic functions of embedded systems;
ii. Discuss the basic structure of embedded systems;
iii. Explain basic concepts of embedded systems;
iv. Discuss the applications of Embedded Systems;
v. Discuss the approaches to the development of embedded software.

Course learning outcomes:

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

i. design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability;
ii. apply and employ specialized knowledge of subsystems like processor cores and other hardware/software system components to design an embedded computer system;
iii. learn and use tools facilitate hardware software simulation and design of embedded computer systems;
iv. demonstrate project report by following a standard IEEE like format, where all the reports are evaluated based on their completeness, English, and citations;
v. display a basic understanding of the issues in managing the project implementation during the design phase of the embedded computer system involving project requirements, specification, simulation, design and prototyping.

Topics:

1.	Introduction to Embedded and Real time Systems	**4 Hours**
2.	Introduction to Hardware Software Co-design	**4 Hours**
3.	Embedded SoPC (System on Programmable Chips)	**4 Hours**
4.	Embedded Processors - ARM Cortex M3, NIOS-II and other CPU Cores	**4 Hours**
5.	Multitasking and Real-time Scheduling Techniques	**4 Hours**

6. Pre-emptive and Non-pre-emptive Scheduling	**4 Hours**
7. ARM CPU, Cortex M3 and Multitasking Application	**6 Hours**
8. RTX – Real-time Operating System	**4 Hours**
9. Real-time Scheduling Techniques.	**4 Hours**
10. Earliest-Deadline-First (EDF) and Rate-Monotonic Scheduling.	**6 Hours**
11. Introduction to Priority Inversion Problem and its Solutions	**6 Hours**
12. Fault-tolerant Embedded Systems	**4 Hours**
13. Hardware and Software Fault-Tolerance Techniques	**6 Hours**
14. Catching up and Review	**4 Hours**
Semester Total Hours	**60 Hours**

Study Material
Laptops with a modern operating system for which the student has administrator privileges, LCD projector, networking tools, printers and fast Internet.

Method of Delivery:
The course will be taught by using lectures, group discussions, seminars, field works and assignments.

Method of Assessment:
Assignments, tests, research assignments and final examination.

Reading Lists:

1. Evans, B. W. (2007). Arduino Programming Note Book. Available online at http://playground.arduino.cc/uploads/Main/arduino_notebook_v1-1.pdf
2. Smith, A.G. (2011). Introduction to Arduino: A piece of cake. Available online at http://www.introtoarduino.com/downloads/IntroArduinoBook.pdf

Course Name:	**Cisco Certified Entry Networking Technology**
Course Code:	**GCS 1302**
Year of Study:	**1**
Semester:	**1**
Contact Hours:	**60**
Credit Units:	**4**

Description:
The course will cover the course content for Cisco Certified Entry Networking Technician (CCENT) curriculum.

Prerequisites: None.

Learning objectives:

i. Install, operate and troubleshoot a small enterprise branch network, including basic network security.
ii. Demonstrates the skills required for entry-level network support positions the starting point for many successful careers in networking.
iii. Design a networks function, identifying major components, function of network components, and the OSI Reading List model.
iv. Design the reasons for connecting networks with routers and how routed networks transmit data through networks using TCP/IP.
v. Design the function of WANs, the major devices of WANs, and configure PPP encapsulation, static and dynamic routing, PAT, and RIP routing.

Course Learning Outcome

i. Using the host-to-host packet delivery process, describe issues related to increasing traffic on an Ethernet LAN and identify switched LAN technology solutions to Ethernet networking issues.
ii. Describe the reasons for extending the reach of a LAN and the methods that can be used, with a focus on RF wireless access.
iii. Use the command-line interface to discover neighbors on the network and manage the router start up and configuration.
iv. Understand how routers and how routed networks transmit data through networks using TCP/IP Design the function of WANs, the major devices of WANs, and configure PPP encapsulation, static and dynamic routing, PAT, and RIP routing.

Topics:

1. Networking Fundamentals:	15 hours
2. Connecting to a WAN:	5 hours
3. Basic Security and Wireless Concepts:	15 hours
4. Routing and Switching Fundamentals:	15 hours
5. TCP/IP and OSI Models:	10 hours
6. WAN Technologies:	5 hours
7. Operating and Configuring IOS Devices:	5 hours
8. Configuring RIPv2, Static and Default Routing:	8 hours
9. Implementing NAT and DHCP:	7 hours
10. Configuring Simple Network:	15 hours
Recess Total Hours	**90 hours**

Study Material

Laptops with a modern operating system for which the student has administrator privileges, LCD projector, networking tools, printers and fast Internet.

Method of Delivery:

The course will be taught by using lectures, group discussions, seminars, field works and assignments.

Method of Assessment:
Assignments, tests, research assignments and final examination. Their relative contributions to the final grade are: course work (Assignments, tests, projects) 30%, final examination 70%, total 100%.

Reading Lists:
1. Due to the volatile nature of the pertinent content, the substantive instructor to access the Reading List materials should guide the student.

Course Name:	**Computer Graphics**
Course Code:	**GCS 2101**
Year of Study:	**2**
Semester:	**1**
Contact Hours:	**45**
Credit Units:	**3**

Description:
Offers an introduction to computer graphics, which has become an increasingly important area within computer science. Computer graphics, particularly in association with the multimedia aspects of the World-Wide Web, have opened up exciting new possibilities for the design of human-computer interfaces. The purpose of this course is to investigate the principles, techniques, and tools that have enabled these advances.

Prerequisites: GCS 1102 Discrete Mathematics; GCS 1202 Introduction to Object Oriented Programming.

Course objectives:

1. Offer a meaningful critique of graphical and multimedia interfaces that incorporates an understanding of the principles of HCI design.
2. Apply the principles that underpin the design of graphics and multimedia systems.
3. Describe the range of tools that can be used to support the development of graphical and multimedia systems.
4. Use existing graphics and multimedia packages to develop appropriate graphical applications.

Course Learning Outcome

1. Design meaningful graphical and multimedia interfaces that allows human-computer interaction
2. Implement the principles that underpin the design of graphics and multimedia systems.
3. Develop and implement appropriate graphical applications using existing graphics and multimedia packages.

Topics covered:

Graphic Systems **2 hours**

Raster and vector graphics systems; video display devices; physical and logical input devices; issues facing the developer of graphical systems.

Fundamental Techniques in Graphics **4 hours**

Hierarchy of graphics software; using a graphics API, simple color models; homogeneous coordinates; affine transformations; viewing transformation; clipping.

Graphical Algorithms **6 hours**

Line generation algorithms; structure and use of fonts; parametric polynomial curves and surfaces; polygonal representation of 3D objects; parametric polynomial curves and surfaces; introduction to ray tracing; image synthesis, sampling techniques, and antialiasing; image enhancement.

Principles of Human-Computer Interaction **10 hours**

Human-centered software development and evaluation.

Graphical User-Interface Design **12 hours**

Choosing interaction styles and interaction techniques; HCI aspects of interface design; dynamics of color; structuring a view for effective understanding.

Graphical User-Interface Programming **12 hours**

Graphical widgets; event management and user interaction; GUI builders and programming environments.

Computer Animation **4 hours**

Key-frame animation; camera animation; scripting system; animation of articulated structures; motion capture; procedural animation; deformation.

Multimedia Techniques **10 hours**

Sound, video, and graphics; design of multimedia systems; tools for multimedia development; virtual reality.

Semester Total **60 Hours**

Study Material

Laptops with a modern operating system for which the student has administrator privileges, LCD projector, networking tools, printers and fast Internet.

Method of Delivery:

The course will be taught by using lectures, practical exercise, laboratory work, projects and assignments.

Method of Assessment:

Assignments, tests, research assignments and final examination. Their relative contributions to the final grade are: course work (Assignments, tests, projects) 30%, final examination 70%, total 100%.

Textbook(s) and other required material:

1. Peter Shirley (2009). Fundamentals of Computer Graphics. 3rd ed., A K Peters.
2. Dave Shreiner and The Khronos OpenGL ARBWorking Group (2009). OpenGL Programming Guide. 7th ed., Addison-Wesley.

Course Name:	**Data Structures, Algorithms and Analysis**
Course Code:	**GCS 2102**
Year of Study:	**2**
Semester:	**1**
Contact Hours:	**60**
Credit Units:	**4**

Description:
Builds on the foundation provided by the GCS 1104 and 1202 sequence to introduce the fundamental concepts of data structures and the algorithms that proceed from them. Topics include recursion, the underlying philosophy of object-oriented programming, fundamental data structures (including stacks, queues, linked lists, hash tables, trees, and graphs), the basics of algorithmic analysis, and an introduction to the principles of language translation.

Prerequisites: GCS 1102 Discrete Mathematics; GCS 1202 Introduction to Object Oriented Programming.

Course objectives:

1. Use standard analysis and design techniques to produce a team-developed, medium-sized, secure software application that is fully implemented and formally tested.
2. Compare and contrast a range of searching and sorting algorithms and analyze time and space efficiencies.
3. Assess the appropriateness of using recursion to solve a given problem.

Course Learning Outcome

1. Design and construct programming solutions using a variety of recursive techniques.
2. Analyze the efficiency of recursive algorithms.
3. Design and develop reusable software using appropriate data structures and templates.
4. Create effective, efficient and secure software, reflecting standard principles of software engineering and software assurance.

Topics covered:
Review of elementary programming concepts. **3 hours**

Fundamental Data Structures: 13 hours

Stacks; queues; linked lists; hash tables; trees; graphs.

Object-Oriented Programming: 8 hours

Object-oriented design; encapsulation and information hiding; classes; separation of behavior and implementation; class hierarchies; inheritance; polymorphism.

Fundamental Computing Algorithms: 10 hours

O(N log N) sorting algorithms; hash tables, including collision avoidance strategies; binary search trees; representations of graphs; depth-first and breadth-first traversals.

Recursion: 5 hours

The concept of recursion; recursive mathematical functions; simple recursive procedures; divide-and-conquer strategies; recursive backtracking; implementation of recursion.

Basic Algorithmic Analysis: 3 hours

Asymptotic analysis of upper and average complexity bounds; identifying differences among best, average, and worst case behaviours; Big "O", little "o", omega, and theta notation; standard complexity classes; empirical measurements of performance; time and space trade-offs in algorithms; using recurrence relations to analyze recursive algorithms.

Algorithmic Strategies: 5 hours

Brute-force algorithms; greedy algorithms; divide-and-conquer; backtracking; branch-and-bound; heuristics; pattern matching and string/text algorithms; numerical approximation algorithms.

Graphs and Trees: 4 hours

Trees; undirected graphs; directed graphs; spanning trees/forests; traversal strategies.

Basic Computability: 5 hours

Finite-state machines; context-free grammars; tractable and intractable problems; uncomputable functions; the halting problem; implications of uncomputability.

Overview of Programming Languages: 2 hours

History of programming languages; brief survey of programming paradigms; procedural languages; object-oriented languages; functional languages; declarative, non-algorithmic languages; scripting languages; the effects of scale on programming methodology.

Software Validation: 2 hours

Validation planning; testing fundamentals, including test plan creation and test case generation; black-box and white-box testing techniques; unit, integration, validation, and system testing; Object-oriented testing; inspections.

Semester Total 60 Hours

Study Material

Laptops with a modern operating system for which the student has administrator privileges, LCD projector, networking tools, printers and fast Internet.

Method of Delivery:
The course will be taught by using lectures, group discussions, seminars, field works and assignments.

Method of Assessment:
Assignments, tests, research assignments and final examination. Their relative contributions to the final grade are: course work (Assignments, tests, projects) 30%, final examination 70%, total 100%.

Textbook(s) and other required material:

1. Mark Allen Weiss (2005). Data Structures & Problem Solving using Java. 2nd ed., Addison-Wesley.
2. Kathy Sierra and Bert Bates (2005). Head First Java. 2nd ed., O'Reilly.
3. Michael T. Goodrich and Roberto Tamassia (2006). Data Structures and Algorithms in Java. 4th ed., John Wiley & Sons.
4. Michael T. Goodrich and Roberto Tamassia (2001). Algorithm Design: Foundations, Analysis, and Internet Examples. 1st ed., John Wiley & Sons, Inc.
5. Jon Kleinberg and Eva Tardos (2005). Algorithm Design. Addison Wesley, United States ed edition.
6. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein (2009). Introduction to Algorithms. 3rd ed., The MIT Press.
7. Kurt Mehlhorn and Peter Sanders (2008). Algorithms and Data Structures: The Basic Toolbox. Springer-Verlag Berlin Heidelberg.

Course Name:	**Systems Modeling**
Course Code:	**GCS 2103**
Year of Study:	**2**
Semester:	**1**
Contact Hours:	**45**
Credit Units:	**3**

Description:
This course provide students with an understanding of systems modelling theory and practice, the skills to create system models from informal specifications, and an appreciation of how to map such models to implementations.

Learning objectives:

i. Illustrate the need for systems modelling.
ii. Describe how systems modelling is related to the various stages of software development.
iii. UML for some simple cases.
iv. Perform requirement analysis to create system models from simple requirement specifications written in English.

Course Learning Outcome

i. Explain the fundamental principles of systems modelling;
ii. Understand how system models may be created and expressed in the
iii. Read and express system models in the UML.
iv. Map system models expressed in the UML to program or database implementations.

Topics:

Systems Modelling **8 hours**

The need for modelling; models of the Unified Modeling Language (UML); interaction of selected UML models; validating and verifying models.

Use Case Modelling **6 hours**

Use Case diagrams (concepts and notation) and; Use Case descriptions.

Class Modelling **12 hours**

Objects and classes; classes and attributes; abstraction and encapsulation; associations; generalisation and inheritance.

Interaction Diagrams **7 hours**

Sequence diagrams; classes and operations; polymorphism.

Activity Diagrams **6 hours**

Users and task analysis; activity diagrams; modelling the user interface.

Mapping System Models **6 hours**

Mapping system models into software program or database implementations

Semester Total Hours **45 hours**

Study Material

Laptops with a modern operating system for which the student has administrator privileges, LCD projector, networking tools, printers and fast Internet.

Method of Delivery:

The course will be taught by using lectures, group discussions, seminars, field works and assignments.

Method of Assessment:

Assignments, tests, research assignments and final examination.

Reading Lists:

1. Martin Fowler and Kendall Scott (2003). UML Distilled: A Brief Guide to the Standard Object Modelling Language. 3rd ed., Addison-Wesley Professional.
2. Craig Larman (2004). Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development. 3rd ed., Prentice Hall.
3. Grady Booch, Ivar Jacobson and James Rumbaugh (2005). The Unified Modelling Language User Guide. 2rd ed., Addison-Wesley Professional.

Course Name:	**Computer Organization and Architecture**
Course Code:	**GCS 2104**
Year of Study:	**2**
Semester:	**1**
Contact Hours:	**45**
Credit Units:	**3**

Description:

Introduces students to the organization and architecture of computer systems, beginning with the standard von Neumann model and then moving forward to more recent architectural concepts.

Course Objectives:

i. Introduce Writing and debugging simple programs using assembly code.
ii. Explain the principles underlying the design and development of computer systems for a variety of purposes.
iii. Provide computing developments (such as compiler technology, networking, the web, multimedia, safety, security) on the architecture of computer systems.
iv. Define architectural features of a modern computer system.

Course Learning Outcome

Upon completion of this course students will have the ability to:

i. Write and debug simple programs using assembly code.
ii. Explain the principles underlying the design and development of computer systems for a variety of purposes.
iii. Trace the influences of important computing developments (such as compiler technology, networking, the web, multimedia, safety, security) on the architecture of computer systems.
iv. State the architectural features of a modern computer system.

Topics:

Digital Logic **8 hours**

Fundamental building blocks (logic Gates, flip-flops, counters, registers,
PLA); logic expressions, minimization, sum of product forms; register transfer notation; physical considerations (Gate delays, fan-in, fan-out).

Data Representation 6 hours

Bits, bytes, and words; numeric data representation and number bases; fixed- and floating-point systems; signed and twos complement representations; representation of nonnumeric data (character codes, graphical data); representation of records and arrays.

Assembly Level Organization **7 hours**

Basic organization of the von Neumann machine; control unit; instruction fetch, decode, and execution; instruction sets and types (data manipulation, control, I/O); assembly/machine language programming; instruction formats; addressing modes; subroutine call and return mechanisms; I/O and interrupts.

Memory Systems **5 hours**

Storage systems and their technology; coding, data compression, and data integrity; memory hierarchy; main memory organization and operations; latency, cycle time, bandwidth, and interleaving; cache memories (address mapping, block size, replacement and store policy); virtual memory (page table, TLB); fault handling and reliability.

Interfacing and Communication **6 hours**

I/O fundamentals: handshaking, buffering, programmed I/O, interrupt driven I/O; interrupt structures: vectored and prioritized, interrupt acknowledgment; external storage, physical organization, and drives; buses: bus protocols, arbitration, and direct memory access

(DMA); introduction to networks; multimedia support; raid architectures.

Functional Organization **7 hours**

Implementation of simple data paths; control unit: hardwired realization vs. micro programmed realization; instruction pipelining; introduction to instruction-level **parallelism (ILP).**

Multiprocessor and Alternative Architectures **6 hours**

Introduction to SIMD, MIMD, VLIW, EPIC; systolic architecture; interconnection networks; shared memory systems; cache coherence; memory models and memory **consistency.**

Performance Enhancements **6 hours**

RISC architecture; branch prediction; prefetching scalability.

Contemporary Architectures **3 hours**

Hand-held devices; embedded systems; trends in processor architecture.

Semester Total Hours **45 hours**

Study Material

Laptops with a modern operating system for which the student has administrator privileges, LCD projector, networking tools, printers and fast Internet.

Method of Delivery:

The course will be taught by using lectures, practical exercise, laboratory work, projects and assignments.

Method of Assessment:

Assignments, tests, exercises and final examination.

Their relative contributions to the final grade are: course work (assignments, tests, laboratory, and projects) 30%, final examination 70%, total 100%.

Reading Lists:

1. Linda Null and Julia Lour (2006). The Essentials of Computer Organization and Architecture. 2nd ed., Jones and Barlett Publishers.
2. David A. Patterson and John L. Hennessy (2007). Computer Organization and Design; the Hardware/Software Interface. 3rd ed., Morgan Kaufmann.

3. Sivarama P. Dandamudi (2003). Fundamentals of Computer Organization and Design. Springer.
4. William Stallings (2006). Computer Organization and Architecture: Designing for Performance. 7th ed., Pearson Prentice Hall.

Course Name:	**Organizational Behaviour**
Course Code:	**GCS 2105**
Year of Study:	**2**
Semester:	**1**
Contact Hours:	**45**
Credit Units:	**3**

Description:

This course will provide the student with an overview of the fundamental theories and principles of organizational behaviour and illustrate how these theories are translated into practice within organizations. It will demonstrate the importance of understanding behaviour at the individual, the group, and the organizational levels when managing organizations. An integral part of this course is to develop student's leadership and interpersonal skills through experiential exercises and discussion and to provide students with an opportunity to evaluate their own experiences, preferences, strengths, and weaknesses, and how these may impact their ability to become effective members of organizations. Topics include: employee motivation, attitudes and values, decision making, managing conflict, team development, leadership theory, and corporate and organizational culture.

Prerequisites: None.

Course Objectives:

This course aims to:

i) Introduce students to organizational behaviour theories to identify and analyse issues and problems and solve organizational case studies.
ii) Enable students to synthesize information to make decisions and solve problems using organizational behaviour concepts, theories and principles.
iii) Support students to analyse the impact of individuals and team behaviour on organizational productivity.
iv) Enable students to evaluate the impact of organizational structure, design, culture and change
v) Support students to work effectively individually and in groups.

Course Learning Outcomes:

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

i) Apply organizational behaviour theories to identify and analyse issues and problems and solve organizational case studies.
ii) Synthesize information to make decisions and solve problems using organizational behaviour concepts, theories and principles.
iii) Analyse the impact of individuals and team behaviour on organizational productivity.
iv) Evaluate the impact of organizational structure, design, culture and change
v) Demonstrate working effectively individually and in groups.

Detailed Course Content:
Introduction to Organizational Behavior **2 Hours**
What is organization behavior (OB); impact on business and strategic management; analysis of open systems; current OB challenges/opportunities?

Attitudes, Values and Ethics: **2 Hours**
Job satisfaction, organizational commitment, attitudes, behaviors and outcomes; values.

Personality and Perception: **2 Hours**
Personality theories; organizational characteristics.

Motivation: **2 Hours**
Motivational theories.

Job Design: **2 Hours**
What is it and why it is important; approaches to job design; job characteristic model.

Communication: **4 Hours**
Communication process model, barriers to communication, effective interpersonal communication.

Team Dynamics and Effectiveness: **4 Hours**
How teams develop; team effectiveness; team roles.

Creativity and Decision Making: **4 Hours**
Creative process; Organizational influences on creativity; decision making models.

Power and Influence: **6 Hours**
Sources and effects of power; organizational power enablers; converting power to influence.

Leadership: **4 Hours**
Leadership theories and applications.

Conflict: **4 Hours**
Nature of organizational conflict. Sources of conflict. Preventing and managing conflict.

Organizational Culture: **6 Hours**
Elements of culture; culture and performance; leadership and culture; changing culture.

Organizational Structure and Organizational Change: **3 Hours**

Semester Total Hours **45 Hours**

Study Material
Laptops with a modern operating system for which the student has administrator privileges, a server computer, LCD projector and fast Internet

Mode of Delivery

Lectures, tutorials and instructor-led seminars will be the main method of instruction. In addition, practical exercises for both the group and individuals will be given based on real life case studies.

Mode of Assessment

The assessment will be in form of tests (15%) and class presentations (5%) individual assignment (10%) and end of semester examinations (70%).

Reading List:

1. Bratton, J., Sawchuk, P., Forshaw, C., Callinan, & Corbett (2010) Work And Organisational Behaviour, 2nd Ed., Palgrave.
2. Huczynski, A. and Buchanan, DA. (2013) Organisational Behaviour, 8th Ed., Pearson.
3. McShane, S. L., Steen, S. L., & Tasa, K. (2015). *Canadian organizational behaviour* (9th ed.). Toronto: McGraw-Hill Ryerson
4. Mcshane, SL., Olekalns, M. & Travaglione, T. (2010) Organisational Behaviour on the Pacific Rim, 3rd ed., McGraw-Hill.
5. Pat R. Sniderman, Julie Bulmash, Debra L. Nelson and James Campbell Quick (2010). *Managing Organizational Behaviour in Canada*, 2nd ed., Thomson Nelson.
6. Stephen P. Robbins and Timothy A. Judge (2010). *Student Value Edition for organizational Behavior*. 14th ed., Prentice Hall.

Course Name:	**Databases**
Course Code:	**GCS 2106**
Year of Study:	**2**
Semester:	**1**
Contact Hours:	**60**
Credit Units:	**4**

Description:

Introduces the concepts and techniques of database systems.

Course objectives:

This course unite aim student to:

i. Define approaches to ensure that information systems can scale from the individual to the global. Identify issues of data persistence to an organization.
ii. Define basic goals, functions, models, components, applications, and social impact of database systems, categorize data models based on the types of concepts that they provide to describe the database structure that is, conceptual data model, physical data model, and representational data model.
iii. Provide modelling concepts and notation of the entity-relationship model and UML, including their use in data modelling, Prepare a relational schema from a conceptual model developed using the entity- relationship model.
iv. Define concepts of entity integrity constraint and referential integrity constraint (including definition of the concept of a foreign key).

v. Develop relational algebra operations from mathematical set theory (union, intersection, difference, and Cartesian product) and the relational algebra operations developed specifically for relational databases (select (restrict), project, join, and division), set of query processing strategies and select the optimal strategy.
vi. Design functional dependency between two or more attributes that are a subset of a relation.

Expected outcomes:
By the end of this course unit student should be able to

i. Explain and demonstrate the concepts of entity integrity constraint and referential integrity constraint (including definition of the concept of a foreign key).
ii. Demonstrate use of the relational algebra operations from mathematical set theory (union, intersection, difference, and Cartesian product) and the relational algebra operations developed specifically for relational databases (select (restrict), project, join, and division), set of query processing strategies and select the optimal strategy.
iii. Determine the functional dependency between two or more attributes that are a subset of a relation.
iv. Describe approaches to ensure that information systems can scale from the individual to the global. Identify issues of data persistence to an organization.
v. Cite the basic goals, functions, models, components, applications, and social impact of database systems, categorize data models based on the types of concepts that they provide to describe the database structure that is, conceptual data model, physical data model, and representational data model.
vi. Describe the modelling concepts and notation of the entity-relationship model and UML, including their use in data modelling, Prepare a relational schema from a conceptual model developed using the entity- relationship model.

Topics:

Information Models and Systems 5 hours

History and motivation for information systems; information storage and retrieval; information management applications; information capture and representation; analysis and indexing; search, retrieval, linking, navigation; information privacy, integrity, security, and preservation; scalability, efficiency, and effectiveness; concept of information assurance (data persistence, integrity).

Database Systems 5 hours

History and motivation for database systems; components of database systems; DBMS functions; database architecture and data independence.

Data modeling 8 hours

Data modeling; conceptual models; object-oriented model; relational data model.

Relational Databases 10 hours

Mapping conceptual schema to a relational schema; entity and referential integrity; relational algebra and relational calculus.

Database Query Languages 8 hours

Overview of database languages; SQL; query optimization; 4thgeneration environments; embedding non-procedural queries in a procedural language; introduction to Object Query language.

Relational Database Design 8 hours

Database design; functional dependency; formal forms; multivalued dependency; join dependency; representation theory.

Transaction Processing 7 hours

Transactions; failure and recovery; concurrency control.

Distributed Databases 6 hours

Distributed data storage; distributed query processing; distributed transaction model; concurrency control; homogeneous and heterogeneous solutions; client-server.

Physical Database Design 5 hours

Storage and file structure; indexed files; hashed files; signature files; b-trees; files with dense index; files with variable length records; database efficiency and tuning.

Semester Total Hours 60 hours

Study Material

Laptops with a modern operating system for which the student has administrator privileges, LCD projector, networking tools, printers and fast Internet.

Method of Delivery:

The course will be taught by using lectures, group discussions, seminars, field works and assignments.

Method of Assessment:
Assignments, tests, research assignments and final examination.

Reading Lists:

1. C. J. Date (2003). An Introduction to Database Systems. 8th ed., Addison Wesley.
2. Hector Garcia-Molina, Jeff Ullman, and Jennifer Widom (2008). Database Systems: The Complete Book. 2nd ed., Prentice Hall.
3. Jeff Ullman, and JenniferWidom (2007). A First Course in Database Systems. 3rd ed., Prentice Hall.
4. R. El Masri and S.B. Navathe (2007). Fundamentals of Database Systems. 5th ed., Addison Wesley.

5. T. Connolly, and C. Begg (2014). Database Systems: A Practical Approach to Design, Implementation and Management. 6th ed., Pearson.

Course Name:	**Automata, Computability and Complexity**
Course Code:	**GCS 2201**
Year of Study:	**2**
Semester:	**2**
Contact Hours:	**45**
Credit Units:	**3**

Course Description:
This course introduces basic mathematical models of computation and finite representation of infinite objects. Topics covered include: finite automata and regular languages, context-free languages, Turing machines, partial recursive functions, Church's Thesis, undecidability, reducibility and completeness, time complexity and NP-completeness, probabilistic computation, and interactive proof systems.

Prerequisites: GCS 1102 Discrete Mathematics

Course Objectives

1. Explain the theoretical limits on computational solutions of undecidable and inherently complex problem.
2. Describe concrete examples of computationally undecidable or inherently infeasible problems from different fields.
3. Device and analyze the complexity of procedures to determine properties of computationally bounded automata.
4. Understand and explain formal definitions of machine models.
5. Prove the undecidability or complexity of a variety of problems.

Course Learning Outcome:
Upon completion of this course students will have the ability to have:

1. Understanding of Turing-recognizable languages, Turing-computable functions, and the difference between solvable and unsolvable problems
2. Ability to prove unsolvability by reduction

3. Understanding the time and space complexity of Turing machines, the complexity classes P and NP, and NP-completeness
4. Ability to prove NP-completeness by reduction

Topics:

Finite Automata, Regular Languages and Expressions: **14 hours**

Deterministic finite automata (DFAs); nondeterministic finite automata (NFAs); regular expressions; non-regular languages, algorithms for automata.

Computability Theory: **26 hours**

Turing machines; nondeterministic Turing machines; undecidability; PCP; counter and stack machines; reducibility; recursion theorem.

Complexity Theory: **20 hours**

Time complexity; nondeterministic time complexity; NP-completeness; Cook-Levin theorem; advanced time complexity; space complexity; probabilistic complexity; probabilistic complexity and interactive proofs.

Semester Total Hours **60 hours**

Study Material

Laptops with a modern operating system for which the student has administrator privileges, LCD projector, networking tools, printers and fast Internet.

Method of Delivery:

The course will be taught by using lectures, case studies, and assessments.

Method of Assessment:

Assessments, tests, and final examination. Their relative contributions to the final grade are: course work (assignments, tests, laboratory, projects) 30%, final examination 70%, total 100%.

Textbooks:

1. Sipser Michael (2005). Introduction to the Theory of Computation. 2nd ed. Boston, MA; Course Technology. ISBN: 0534950973.
2. Martin John (2002). Introduction to Languages and the Theory of Computation. New York, NY: McGraw Hill. ISBN: 0072322004.
3. Kozen Dexter (1999). Automata and Computability. New York, NY: Springer-Verlag. ISBN: 0387949070.
5. Garey Michael and David S. Johnson (1979). Computers and Intractability: A Guide to the Theory of NP-Completeness. New York, NY: W.H. Freeman and Company. ISBN: 0716710455.

Course Name: **Operating Systems**
Course Code: **GCS 2202**
Year of Study: **2**
Semester: **2**

Contact Hours: **45**
Credit Units: **3**

Description:
Introduces the fundamentals of operating systems design and implementation. Topics include an overview of the components of an operating system, mutual exclusion and synchronization, implementation of processes, scheduling algorithms, memory management, and file systems.

Prerequisite: GCS 2104 Computer Organization and Architecture

Course objectives:
The main aim of this course unit is to:

1. Explain the objectives and functions of modern operating systems.
2. Understand theoretical concepts and programming constructs used for operation of modern operating systems.
3. Gain practical experience with software tools available in modern operating systems such as semaphores, system calls, sockets and threads.
4. Describe and explain how computing resources are used by application software and managed by system software.
5. Describe and explain the need for concurrency within the framework of an operating system.
6. Describe and explain relationships between scheduling algorithms and application domains.

Course Learning Outcomes
By the end of course unit students should be able to:

1. Understand the objectives and functions of modern operating systems.
2. Apply theoretical concepts and programming constructs used for operation of modern operating systems.
3. Use practical software tools available in modern operating systems such as semaphores, system calls, sockets and threads.
4. Use computing resources for application software and system software.
5. Apply concurrency within the framework of an operating system.
6. Use and explain relationships between scheduling algorithms and application domains.

Topics

Overview **3 hours**
Role and purpose of operating systems; history of operating system development; functionality of a typical operating system; design issues (efficiency, robustness, flexibility, portability, security, compatibility).

Operating System Principles **3 hours**
Structuring methods; abstractions, processes, and resources; design of application programming interfaces (APIs); device organization; interrupts; concept of user/system state and protection, transition to kernel mode.

Concurrency **6 hours**

The idea of concurrent execution; states and state diagrams; implementation structures (ready lists, process control blocks, and so forth); dispatching and context switching; interrupt handling in a concurrent environment.

Mutual Exclusion **6 hours**
Definition of the \mutual exclusion" problem; deadlock detection and prevention; solution strategies; models and mechanisms (semaphores, monitors, condition variables, rendezvous); producer consumer problems; synchronization; multiprocessor issues.

Scheduling and Dispatch **6 hours**
Preemptive and non preemptive scheduling; scheduling policies; processes and threads; deaslines real-time issues.

Memory Management **6 hours**
Review of physical memory and memory management hardware; overlays, swapping, and partitions; paging and segmentation; page placement and replacement policies; working sets and thrashing; caching.

Device Management **3 hours**
Characteristics of serial and parallel devices; abstracting device differences; buffering strategies; direct memory access; recovery from failures.

File Systems **6 hours**
Files (data, metadata, operations, organization, buffering, sequential, non-sequential); directories (contents and structure); file systems (partitioning, mount/unmount, virtual file systems); standard implementation techniques; memory-mapped files; special purpose file systems; naming, searching, access, backups.

Security and Protection **6 hours**
Overview of system security; policy/mechanism separation; security methods and devices; protection, access, and authentication; models of protection; memory protection; encryption; recovery management.

Semester Total Hours **45 hours**

Study Material
Laptops with a modern operating system for which the student has administrator privileges, LCD projector, networking tools, printers and fast Internet.

Method of Delivery:
The course will be taught by using lectures, practical exercise, laboratory work, case-studies and assignments.

Method of Assessment:
Assignments, tests, exercises and final examination.
Their relative contributions to the final grade are: course work (assignments, tests, laboratory, and projects) 30%, final examination 70%, total 100%.

Reading lists:

1. Abraham Silberschatz, Peter Galvin and Greg Gagne (2010). Operating System Concepts. 8th ed., John Wiley and Sons Inc.
2. H.M. Deitel, P.J. Deitel and D.R. Ghoffnes (2003). Operating Systems. 3rd ed., Pearson Prentice Hall Ltd., 2003.
3. Andrew S. Tanenbaum and Albert S (2006). Woodhull. Operating Systems Design and Implementation. 3rd ed., Pearson Prentice Hall Ltd.

Course Name:	**Social & Professional Issues**
Course Code:	**GCS 2203**
Year of Study:	**2**
Semester:	**2**
Contact Hours:	**45**
Credit Units:	**3**

Description:
Introduces students to the social and professional issues that arise in the context of computing.

Course Objective

i. Understand the issues of morality and technology at the Internet age.
ii. Have an idea on the development of computer ethics.
iii. Articulate the social trade-offs due to the rise of use of computers.
iv. Know about the ethical problems which are raised at the Internet environment.
v. Identify the different types of computer crimes, and the ethical arguments of privacy protection versus freedom of speech.
vi. Understand the strengths and weaknesses of alternative approaches to protect privacy, intellectual property, freedom of speech, and protection against computer malfunction (hardware and software) and crime.
vii. Distinguish among patent, copyright, and trade secret protections.
viii. Address differences between local, regional, and international copyright laws.
ix. Identify ethical issues that arise in a software development environment and how to address them.
x. Understand the concept of professional ethics, and analyze an argument to identify premises and conclusion (case studies).
xi. Use examples, analogy, and counter-analogy in ethical arguments, and articulate the ethical trade-offs in a technical decision.

Course Learning Outcomes

On completion of this course unit, the students will be able to:

1. Apply the basic cultural, social, legal, and ethical issues inherent in the discipline of computing.
2. Trace the history of IT discipline, where it is, and where it is heading.

3. Carry out their individual roles in this process, as well as appreciate the philosophical questions, technical problems, and aesthetic values that play an important part in the development of the discipline.
4. Ask serious questions about the social impact of computing and to evaluate proposed answers to those questions.
5. Apply the basic legal rights of software and hardware vendors and users, and also appreciate the ethical values that are the basis for those rights.

Topics

History of Computing **3 hours**

Prehistory the world before 1946; history of computer hardware, software, networking; pioneers of computing.

Social Context **6 hours**

Introduction to the social implications of computing; social implications of networked communication; growth of, control of, and access to the Internet; gender-related issues; international issues.

Analytical Tools **6 hours**

Making and evaluating ethical arguments; identifying and evaluating ethical choices; understanding the social context of design; identifying assumptions and values.

Professional Ethics **6 hours**

Community values and the laws by which we live; the nature of professionalism; various forms of professional credentialing and the advantages and disadvantages; the role of the professional in public policy; maintaining awareness of consequences; ethical dissent and whistle-blowing; codes of ethics, conduct, and practice; dealing with harassment and discrimination; “Acceptable use" policies for computing in the workplace.

Risks **6 hours**

Historical examples of software risks; implications of software complexity; risk assessment and management.

Security Operations **3 hours**

Physical security; physical access controls; personnel access controls; operational security; security policies for systems/networks; recovery and response; dealing with problems (both technical and human).

Intellectual Property **6 hours**

Foundations of intellectual property; copyrights, patents, and trade secrets; software piracy; software patents; transnational issues concerning intellectual property.

Privacy and Civil Liberties **6 hours**

Ethical and legal basis for privacy protection; privacy implications of massive database systems; technological strategies for privacy protection; freedom of expression in cyberspace; international and intercultural implications.

Computer Crime **3 hours**

History and examples of computer crime; “cracking” and its effects; viruses, worms, and Trojan horses; crime prevention strategies.

Economics of Computing **3 hours**

Monopolies and their economic implications; effect of skilled labor supply and demand on the quality of computing products; pricing strategies in the computing domain; differences in access to computing resources and the possible effects thereof.

Philosophical Frameworks **3 hours**

Philosophical frameworks, particularly utilitarianism and deontological theories; problems of ethical relativism; scientific ethics in historical perspective; differences in scientific and philosophical approaches.

Semester Total **45 hours**

Study Material

Laptops with a modern operating system for which the student has administrator privileges, LCD projector, networking tools, printers and fast Internet.

Method of Delivery:

The course will be taught by using lectures and assignments (group essay, individual essay and group presentation).

Method of Assessment:

Assignments, tests, and final examination. Their relative contributions to the final grade are: course work (assignments, tests, laboratory, and projects) 30%, final examination 70%, total 100%.

Reading Lists

1. Quinn J. Michael (2010). Ethics for the Information Age. 4th ed., Addison Wesley.
2. Penny Duquenoy, Simon Jones and Barry G. Blundell (2007). Ethical, Legal and Professional Issues in Computing. 1st ed., Cengage Learning Business Press.
3. Frank Bott (2005). Professional Issues in Information Technology. British Computer Society.
4. Robert Ayres (1998). Essence of Professional Issues in Computing. 1st ed., Prentice Hall.
5. D.I. Bainbridge. Introduction to Information Technology Law. Pearson.

Course Name:	**Software Engineering**
Course Code:	**GCS 2204**
Year of Study:	**2**
Semester:	**2**
Contact Hours:	**60**
Credit Units:	**4**

Description:

Software engineering is the discipline concerned with the application of theory, knowledge, and practice for effectively and efficiently building software systems that satisfy the requirements of users and customers.
This course covers fundamentals of software engineering: software design, using APIs, software tools and environments, software processes, software requirements and specifications, software validation and evolution, and software project management.

Prerequisites: GCS 1102 Discrete Mathematics, GCS 1202 Introduction to Object Oriented Programming and GCS 2103 Systems Modeling.

Course Objectives
This module aims to:

i. Provide students comprehensive details to introduction to software engineering.
ii. Introduce the concepts and methods required for the construction of large software intensive systems. It aims to develop a broad understanding of the discipline of software engineering.
iii. Provide a detailed knowledge of techniques for the analysis and design of complex software intensive systems. It aims to set these techniques in an appropriate engineering and management context.
iv. Give an introduction to basic concepts, principles and techniques used in software engineering.
v. Discuss the nature of software and software projects, review of object orientation, software development on reusable technology, developing requirements, modelling with classes, design patterns, focusing on users and their tasks, modelling interactions and behaviour, testing and managing software process.

Course Learning Outcomes
On completion of this course unit, the students will be able to:

i). Understand what software engineering is and why it is important;
ii). Apply the concept of software processes and software process models;
iii). Use the principals of object orientation in software engineering;
iv). Adopt the principals of software development on reusable technology;
v). Apply the various types of software requirements (Functional & Noun Functional) in software development;
vi). Implement and accomplish effective requirements management through an effective software team.

Topics
Software Design **8 hours**
Fundamental design concepts and principles; the role and the use of contracts; design patterns; software architecture; structured design; object-oriented analysis and design; component-level design; design qualities; internal including low coupling, high cohesion, information hiding, efficiency; external including reliability, maintainability, usability, performance; other approaches: data structured centered, aspect oriented, function oriented, service oriented, agile; design for reuse; use of open-source materials.

Using APIs **7 hours**

Programming using APIs; design of APIs; class browsers and related tools; debugging in the API environment Introduction to component-based Entercomputing; introduction to component-based computing.

Tools and Environments **6 hours**

Programming environments; requirements analysis and design modeling tools; testing tools including static and dynamic analysis tools; tools for source control, and their use in particular in team-work; configuration management and version control tools; tool integration mechanisms.

Software Processes **7 hours**

Software life-cycle and process models; software process capability maturity models; approaches to process improvement; process assessment models; software process measurements.

Requirements Specifications **8 hours**

Systems level considerations; software requirements elicitation; requirements analysis modeling techniques; functional and non-functional requirements; acceptability of certainty / uncertainty considerations regarding software / system behaviour; prototyping; basic concepts of formal specification techniques.

Software Verification and Validation **6 hours**

Distinguishing between verification and validation; static approaches and dynamic approaches; validation planning; documentation for validation; different kinds of testing - human computer interface, usability, reliability, security, conformance to specification; testing fundamentals, including test plan creation and test case generation black-box and white-box testing techniques; defect seeding; unit, integration, validation, and system testing; object-oriented testing; systems testing; measurements: process, design, program; verification and validation of non-code (documentation, help _les, training materials); fault logging, fault tracking and technical support for such activities; regression testing; inspections, reviews, audits.

Software Evolution **6 hours**

Software maintenance; characteristics of maintainable software; reengineering legacy systems; refactoring; software reuse.

Software Project Management **6 hours**

Team management - team processes, team organization and decision making, roles and responsibilities in a software team, role identification and assignment, project tracking, team problem resolution; project scheduling; software measurement and estimation techniques; risk analysis - the issue of security, high integrity systems, safety critical systems, the role of risk in the life cycle; software quality assurance - the role of measurements; software configuration management and version control; release management; project management tools; software process models and process measurements.

Formal Methods **6 hours**

Formal methods concepts; formal specification languages; model checking; executable and non-executable specifications; pre and post assertions; formal verification; tools in support of formal methods.

Semester Total **60 hours**

Study Material
Laptops with a modern operating system for which the student has administrator privileges, LCD projector, networking tools, printers and fast Internet.

Method of Delivery:
The course will be taught by using lectures, projects, case-studies and assignments (group presentation).

Method of Assessment:
Assignments, tests, and final examination. Their relative contributions to the final grade are: course work (assignments, tests, laboratory, and projects) 30%, final examination 70%, total 100%.

Reading Lists

1. 1. Roger Pressman (2009). Software Engineering: A Practitioner's Approach. 7th ed., McGraw-Hill Science/Engineering/Math.
2. Ian Sommerville (2010). Software Engineering. 9th ed., AddisonWesley.

Course Name:	**Artificial Intelligence**
Course Code:	**GCS 2205**
Year of Study:	**2**
Semester:	**2**
Contact Hours:	**45**
Credit Units:	**3**

Description:
An introduction to the basic principles, techniques, and applications of Artificial Intelligence. Coverage includes knowledge representation, logic, inference, problem solving, search algorithms, game theory, perception, learning, planning, and agent design. Students will experience programming in AI language tools. Potential areas of further exploration include expert systems, neural networks, fuzzy logic, robotics, natural language processing, and computer vision.

Course Objective
The main objectives of this course include:

1. Gain a historical perspective of AI and its foundations.
2. Become familiar with basic principles of AI toward problem solving, inference, perception, knowledge representation, and learning.
3. Investigate applications of AI techniques in intelligent agents, expert systems, artificial neural networks and other machine learning models.
4. Experience AI development tools such as an ‘AI language’, expert system shell, and/or data mining tool.
5. Experiment with a machine learning model for simulation and analysis.
6. Explore the current scope, potential, limitations, and implications of intelligent systems.

Course Learning Outcomes

On completion of this course unit, the students will be able to:

1. Demonstrate fundamental understanding of the history of artificial intelligence (AI) and its foundations.
2. Apply basic principles of AI in solutions that require problem solving, inference, perception, knowledge representation, and learning.
3. Demonstrate awareness and a fundamental understanding of various applications of AI techniques in intelligent agents, expert systems, artificial neural networks and other machine learning models.
4. Demonstrate proficiency developing applications in an 'AI language', expert system shell, or data mining tool.
5. Demonstrate proficiency in applying scientific method to models of machine learning.
6. Demonstrate an ability to share in discussions of AI, its current scope and limitations, and societal implications.

Topics

Fundamental Issues in Intelligent Systems **2 hours**

History of artificial intelligence; philosophical questions; fundamental definitions; modeling the world; the role of heuristics.

Search and Constraint Satisfaction **7 hours**

Problem spaces; Brute-force search; Best-first search; Two-player games; constraint satisfaction.

Knowledge Representation and Reasoning **5 hours**

Review of propositional and predicate logic; resolution and theorem proving; non-monotonic inference; probabilistic reasoning; Bayes theorem.

Advanced Search **8 hours**

Genetic algorithms; simulated annealing; Local search.

Advanced Knowledge Representation and Reasoning **7 hours**

Structured representation; non-monotonic reasoning; reasoning on action and change; temporal and spatial reasoning; uncertainty; knowledge representation for diagnosis, qualitative representation.

Agents **3 hours**

Definition of agents; successful applications and state-of-the-art agent-based systems; software agents, personal assistants, and information access; multi-agent systems.

Machine Learning and Neural Networks **7 hours**

Definition and examples of machine learning; supervised learning; unsupervised learning; reinforcement learning; introduction to neural networks.

AI Planning Systems **6 hours**

Definition and examples of planning systems; planning as search; operator-based planning; propositional planning.

Semester Total	**45 hours**

Study Material
Laptops with a modern operating system for which the student has administrator privileges, LCD projector, networking tools, printers and fast Internet.

Method of Delivery:
The course will be taught by using lectures, projects, practical, case-studies and assignments (group presentation).

Method of Assessment:
Assignments, tests, and final examination. Their relative contributions to the final grade are : course work (assignments, tests, laboratory, projects) 30%, final examination 70%, total 100%.

Reading Lists

1. Stuart Russell and Peter Norvig (2003). Artificial Intelligence: A Modern Approach. 2nd ed., Prentice Hall.
2. Ivan Bratko (2001). Prolog Programming for Artificial Intelligence. 3rd ed., Addison-Wesley.
3. George F. Luger (2008). Artificial Intelligence: Structures and Strategies for Complex Problem Solving. 6th ed., Addison-Wesley.
4. Jeff Hawkins (2005). On Intelligence. 5th ed., St. Martin's Griffin.

Course Name:	**Programming Project**
Course Code:	**GCS 2206**
Year of Study:	**2**
Semester:	**2**
Contact Hours:	**60**
Credit Units:	**4**

Description:
The aim of the programming project is to give each student the opportunity to show individual creativity and originality, apply where appropriate, knowledge and skills taught in the previous semesters, demonstrate programming and investigative skills, and learn how to undertake a project, using their project management, systems modelling and report writing skills.

Prerequisites: GCS 1202 Introduction to Object Oriented Programming, GCS 1205 Project Management, GCS 2103 Systems Modelling.

Course Objective

1. Appreciate the need for the following in a programming project specification; system design and modelling skills; planning, organisation, analysis and design methods; report presentation and documentation skills.
2. Undertake and use project management and systems modelling techniques in the design, planning and organisation of a project.

Course Learning Outcomes
On completion of this course unit, the students will be able to:

1. Document the design and implementation of software, test, and debug simple programs in an object oriented programming language.
2. Perform and plan elementary software testing.

Topics

Communication Skills	12 hours
Programming	30 hours
Project Management	9 hours
Systems Modeling	9 hours
Semester Total Hours	**60 hours**

Study Material
Laptops with a modern operating system for which the student has administrator privileges, LCD projector, networking tools, printers and fast Internet.

Method of Delivery:
At the start of the semester students are provided with a project outline and given dates for submission of an initial project plan, demonstration of software and submission of the project report. Students will be provided with support classes in the form of compulsory tutorials/lectures and a number of optional help sessions.

Method of Assessment:
Projects and report. Their relative contributions to the final grade are: course work (projects, artifacts, software demonstration) 30%, final project report and project demonstration 70%, total 100%.

Reading Lists

1. References from GCS 1102 Communication Skills.
2. References from GCS 1104 Fundamentals of Programming.
3. References from GCS 1202 Introduction to Object Oriented Programming.
4. References from GCS 1205 Project Management.
5. References from GCS 2103 Systems Modeling using UML.

Course Name: **Industrial Training**

Course Code:	**GCS 2301**
Year of Study:	**2**
Semester:	**3**
Contact Hours:	**75**
Credit Units:	**5**

Description:
The Industrial Training program shall be carried out in an industrial setting. The program shall run for ten weeks during which the student is supervised by a member of the teaching staff of the department. The student prepares a report discussing the training environment, lessons learn, challenges faced and recommendations. This report has to be approved by both the training officer at the training firm and the student's supervisor and should be handed in to the said concerned parties before the beginning of the academic year.

Prerequisites: GCS 1301 Embedded Systems Development, GCS 1302 Cisco Certified Entry Networking Technician (CCENT), GCS 2105 Organizational Behavior and GCS 2206 Programming Project.

Learning objectives:
1. Communicate effectively with fellow workers and supervisors in issues related to projects undertaken.
2. Demonstrate and practice good working ethics and to internalize excellence.
3. Attest and practice high-quality organizational skills in enhancing individual and group effectiveness and productivity.

Course learning outcome
1. Demonstrate creativity and innovation in solving problems related to real-life projects.
2. Exhibit pleasant interpersonal skills in developing understanding and appreciation of individual differences and interpersonal skills in building self-confidence.
3. Work independently or under very minimal supervision.
4. Demonstrate good planning, good management, constant monitoring and quality delivery of project undertaken.

Topics:
Discipline Areas to be covered: **75 hours**
Software testing; application development; project support; database management; IT architecture; user and infrastructure support; networking; Web development; marketing; CRM systems; E-Commerce.

Recess Total Hours **75 hours**

Study Material
Laptops with a modern operating system for which the student has administrator privileges, LCD projector, networking tools, printers and fast Internet.

Method of Delivery:
The cause will be delivered by use of case-studies, projects and report.

Method of Assessment:
Projects and report. Their relative contributions to the final grade are: course work (projects, case-studies, report) 30%, final industrial training report 70%, total 100%.

Reading Lists:

1. Carolyn C. Wise (2009). The Vault Guide to Top Internships. 2009 Edition, Vault, Inc.
2. Jamie Fedorko and Dwight Allott (2006). The Intern Files: How to Get, Keep, and Make the Most of Your Internship. Gallery; Original edition.
3. Lindsey Pollak (2007). Getting from College to Career: 90 Things to Do Be- fore You Join the Real World. Harper Paperbacks.
4. William D. Coplin (2003). 10 Things Employers Want You to Learn in College: The Know-How You Need to Succeed. Ten Speed Press.

Course Name:	**Programming Language Translation**
Course Code:	**GCS 3101**
Year of Study:	**3**
Semester:	**1**
Contact Hours:	**45**
Credit Units:	**3**

Description:
Introduces the theory and practice of programming language translation. Topics include compiler design, lexical analysis, parsing, symbol tables, declaration and storage management, code generation, and optimization techniques.

Prerequisites: GCS 2102 Data Structures, Algorithms & Analysis, GCS 2104 Computer Organization and Architecture and GCS 2201 Automata, Computability and Complexity.

Learning objectives:
The major aim of the course is to

1. Describe the steps and algorithms used by language translators.
2. Recognize the underlying formal models such as finite state automata, push-down automata and their connection to language definition through regular expressions and grammars.
3. Discuss the effectiveness of optimization techniques.

Course Learning Outcome

By the end of the course, students will be able to:

1. Explain the impact of a separate compilation facility and the existence of program libraries on the compilation process.
2. Explain the concept of an abstract data type, and recognize the importance of typing for abstraction and safety.
3. Differentiate between static and dynamic typing, and between type declarations and type inference.

4. Have knowledge on how to build a compiler, through type checking and intermediate code generation, for high level language.

Topics:

Overview of Programming Languages **3 hours**

History of programming languages; brief survey of programming paradigms; the role of language translation in the programming process.

Fundamental Issues in Language Design **3 hours**

General principles of language design; design goals; typing regimes; data structure models; control structure models; abstraction mechanisms.

Virtual Machines **2 hours**

The concept of a virtual machine; hierarchy of virtual machines; intermediate languages.

Introduction to Language Translation **3 hours**

Comparison of interpreters and compilers; language translation phases; machine-dependent and machine-independent aspects of translation; language translation as a software engineering activity.

Lexical Analysis **3 hours**

Application of regular expressions in lexical scanners; hand-coded vs. automatically-generated scanners; formal definition of tokens; implementation of finite-state automata.

Syntactic Analysis **5 hours**

Formal definition of grammars; BNF and EBNF; bottom-up vs. top-down parsing; tabular vs. recursive-descent parsers; error handling; automatic generation of tabular parsers; symbol table management; the use of tools in support of the translation process.

Models of Execution Control **5 hours**

Order of evaluation of sub expressions; exceptions and exception handling; runtime systems.

Declaration, Modularity, and Storage Management **5 hours**

Declaration models; parameterization mechanisms; type parameterization; mechanisms for sharing and restricting visibility of declarations; garbage collection.

Type Systems **5 hours**

Data type as set of values with set of operations; data types; type checking models; semantic models of user-defined types; parametric polymorphism; subtype polymorphism; type-checking algorithms.

Interpretation **3 hours**

Iterative vs. recursive interpretation; iterative interpretation of intermediate code; recursive interpretation of a parse tree.

Code Generation **3 hours**

Intermediate and object code; intermediate representations; implementation of code generators; code generation by tree walking; context-sensitive translation; register use.

Optimization **3 hours**
Machine-independent optimization; dataflow analysis; loop optimizations; machine-dependent optimization.

Semester Total Hours **45 hours**

Study Material
Laptops with a modern operating system for which the student has administrator privileges, LCD projector, networking tools, printers and fast Internet.

Method of Delivery:
The cause will be delivered by use of case-studies, projects and report.

Method of Assessment:
Projects and report. Their relative contributions to the final grade are: course work (projects, case-studies, report) 30%, final industrial training report 70%, total 100%.

Reading Lists:

1. Alfred V. Aho, Monica S. Lam, Ravi Sethi and Jeffrey D. Ullman (2006). Compilers: Principles, Techniques, and Tools. 2nd Edition. Addison Wesley. ISBN-10: 0321486811.
2. Keith Cooper and Linda Torczon (2003). Engineering a Compiler. 1st Edition. Morgan Kaufmann. ISBN-10: 155860698X.
3. Michael L. Scott (2009). Programming Language Pragmatics. 3rd Edition. Morgan Kaufmann. ISBN-10: 0123745144.

Course Name:	**Enterprise Network Management**
Course Code:	**GCS3102**
Year of Study:	**3**
Semester:	**1**
Contact Hours:	**60**
Credit Units:	**4**

Description:
This course provides a solid basis on the theoretical and practical understanding of data communication and equips students with knowledge of managing enterprise computer/data networks. This caters for enterprise critical aspects like reliability, security and user management.

***Prerequisite**:* GCS 1205 - Introduction to Computer Networks; GCS 1302 – CCENT,

Course Objectives

The aim of this course is the enable students:

i. Besides acquiring the theoretical background in enterprise networking, students will learn to set up, configure and interconnect an IP network in the lab sessions.
ii. Network monitoring and management tools will also be introduced to the students.
iii. The students will also acquire the knowledge to use simulation tool to design an enterprise network and evaluate design alternatives.
iv. Based on the knowledge and skills, the students are to finish a design of an enterprise network to support applications such as electronic mails, centralised database access, and client-server applications.
v. Various issues such as IP addresses assignment, choice of internetworking equipment and network performance will be considered in the network design.
vi. Understand concepts and terminology associated with Simple network management protocol (SNMP) and Telecommunication Management Network (TMN)

Course Learning Outcomes

On completion of this course unit, the students will be able to:

i. Design and develop solutions for enterprise network architecture;
ii. Build and configure small-scale enterprise network;
iii. Analyze and identify potential issues in managing enterprise network; and
iv. Manage and maintain enterprise network infrastructure.
v. Appreciate network management as a typical distributed application

Topics

Open Systems Interconnection (OSI) Management Framework 16 hours

This topic discussions focus on the OSI management functional areas (Fault management, Configuration management, Accounting management, Performance management and Security management) and the reasons for the need for network management.

The Infrastructure for Network Management 8 hours

This topic introduces the following concepts: Meaning of network infrastructure and the constituents of network infrastructure, Managing entity, Managed device, Manager, Agent, MIB, Element Management System (EMS), Network Management Systems (NMS) as well as the Telecommunication Management network Layer (TMN)

Network Operations Center (NOC) 8 hours

This will focus on Operation, Administration, Maintenance, and provisioning (OAM&P) activities in a NOC as well as remote and Web-based network management tools (Focus on Open Source tools) used.

The Internet Network Management Framework 16 hours

This topic discusses network management objects known as MIB objects, a data definition language known as SMI (structure of Management Information), Abstract Syntax Notation One (ASN.1) as a standard and notation that describes rules and structures for representing, encoding, transmitting, and decoding data in telecommunications and computer networking, a protocol (i.e. SNMP) for conveying information and commands between a managing entity and an agent executing on behalf of that entity within a managed network device and the Security and administration capabilities enhancements in SNMP v2 and SNMPv3.

Introduction to traffic engineering and load balancing 12 hours

This topic is concerned with performance optimization of networks, focus will be on The Current IP architecture (OSPF metric Calculation), limitation of SPF algorithm, solutions (including load balancing, MPLs, use of source address,) to the problem of SPF algorithm, limitations of load balancing, reasons for traffic engineering, mechanisms for recovering ISP's *sunk costs* and network resource (focus on link bandwidth) control mechanisms used by ISPs.

Semester Total Hours 60 hours

Study Material

Laptops with a modern operating system for which the student has administrator privileges, LCD projector, networking tools, printers and fast Internet.

Method of Delivery:

The cause will be delivered by use of case-studies, projects and report.

Method of Assessment:

Projects and report. Their relative contributions to the final grade are: course work (projects, case-studies, report) 30%, final industrial training report 70%, total 100%.

Reading Lists

1. Kornel Terplan, McGeaw-Hill, (1992). Effective management of local area networks;
2. W. Stallings (1999): SNMP, SNMPv2, SNMPv3, and RMON 1 and 2, Addison-Wesley.
3. D. Zeltserman (1999): A Practical Guide to SNMPv3 and Network Management, Prentice Hall.
4. D. Perkins and E. McGinnis (2000): Understanding SNMP MIBs, Prentice Hall, 1997
5. The Quarterly Newsletter of SNMP Technology, Comment, and Events Volume 8, Number 1 September.
6. Practice of System and Network administration by Thomas A. Limoncelli
7. Red Hat Linux Network Management Tools (CD-ROM included) (Paperback) By Steven Maxwell.

Course Name:	**IT Project Management**
Course Code:	**GIT 3103**
Year of Study:	**1**
Semester:	**2**
Contact Hours:	**45**
Credit Units:	**3**

Description:

This course discusses the processes, methods, techniques and tools that organizations use to manage their information systems projects. The course covers a systematic methodology for initiating, planning, executing, controlling, and closing projects. This course assumes that project management in the modern organization is a complex team-based activity, where various types of technologies (including project management software as well as software to support group collaboration) are an inherent part of the project management process. This course also acknowledges that project management involves both the use of resources from within the firm, as well as contracted from outside the organization.

Course objectives:

The objectives of this course is to:

i. Introduce students to Initiate, specify, and prioritize information systems projects and to determine various aspects of feasibility of these projects.
ii. Define foundations of project management, including its definition, scope, and the need for project management in the modern organization.
iii. Equip the phases of the project management lifecycle, control projects through information tracking and cost and change control techniques.
iv. Understand project teams, including the fundamentals of leadership and team motivation manage project schedules with appropriate techniques and tools.
v. Define project communication, both internal to the team, and external to other project stakeholders, identify project risk, and the techniques for ensuring project risk is controlled.

Course Learning Outcome

By end of this course the students will be able to:

i. Apply foundations of project management, including its definition, scope, and the need for project management in the modern organization.

ii. Manage project execution, including monitoring project progress and managing project change, and appropriately documenting and communicating project status.
iii. Demonstrate phases of the project management lifecycle Control projects through information tracking and cost and change control techniques, construct projects, including administrative, personnel, and contractual closure.
iv. Apply mechanisms for dealing with legal issues in complex project contexts.
v. Demonstrate project communication, both internal to the team, and external to other project stakeholders, identify project risk, and the techniques for ensuring project risk is controlled.

Topics:

Introduction to project management: 3 hours

Project management terminology; project failures and project successes; unique features of IT projects; what is project management**?**

Project management lifecycle: 3 hours

What is the project management lifecycle?; project management and systems development or acquisition; project management context; technology and techniques to support the project management lifecycle; project management processes.

Managing project teams: 4 hours

What is a project team?; project team planning; motivating term members; leadership, power and conflict in project teams; managing global project teams.

Managing project communication: **5 hours**
Managing project communication; enhancing team communication; using collaboration technologies to enhance team communication.

Managing project scope: **5 hours**
Project initiation; project planning; how organizations choose projects; activities; developing the project charter.

Managing project scheduling: **5 hours**
What is project scheduling?; common problems in project scheduling; techniques for project scheduling.

Managing project resources: **6 hours**
What are resources?; types of resources (human, capital, time); techniques for managing resources.

Managing project quality: **5 hours**
What is project quality?; what are the threats to project quality?; how can we measure project quality?; tools for managing project quality.

Managing project risk: **6 hours**
What is project risk?; what are the threats to project risk?; tools for managing project risk.

Managing project procurement: **5 hours**
Alternatives to systems development; external acquisition; outsourcing; steps in the procurement process; managing the procurement process.

Project execution, control & closure: **7 hours**
Managing project execution; monitoring progress and managing change; documentation and communication; common problems in project execution.

Managing project control & closure: **6 hours**
Obtaining information; cost control; change control; administrative closure; personnel closure; contractual closure; project auditing.

Semester Total Hours **60 hours**

Study Material
Laptops with a modern operating system for which the student has administrator privileges, LCD projector, networking tools, printers and fast Internet.

Method of Delivery:
The course will be taught by using lectures, group discussions, seminars, field works and assignments.

Method of Assessment:
Projects and report. Their relative contributions to the final grade are: course work (projects, case-studies, report) 30%, final industrial training report 70%, total 100%.

Reading Lists:

1. James Cadle and Donald Yeates (2007). Project Management for Information System. 5th ed., Prentice-Hall.
2. Philip Weaver (2003). Success in Your Project: A Guide to Student System Development Projects. Financial Times/Prentice-Hall.
4. David Avison and Guy Fitzgerald (2006). Information Systems Development: Methodologies, Techniques and Tools. 4th ed., McGraw-Hill Higher Eduction.
5. Bob Hughes and Mike Cotterell (2009). Software Project Management. 5th ed., McGraw-Hill Higher Education.

Course Name:	**Introduction to Computer Security**
Course Code:	**GCS3104**
Year of Study:	**3**
Semester:	**1**
Contact Hours:	**45**
Credit Units:	**3**

Course Description:

This course is a survey of the fundamentals of information assurance and computer security. Topics include: security standards, policies and best practices; principles of ethical and professional behavior; regulatory compliance and legal investigations; information assurance; risk management and threat assessment; business continuity and disaster recovery planning; security architecture and design; elements of cryptography; digital forensics; physical (environmental) security; networking fundamentals; access control and authentication; network and application security; exploiting network, web, software and insider vulnerabilities.

Prerequisites: GCS 1202 Introduction to Object Oriented Programming, GCS 2106 Databases and GCS 2202 Operating Systems.

Course Objectives

This course aim students to:

i). State the basic concepts in information security, including security policies, security models, and security mechanisms.

ii). Explain concepts related to applied cryptography, including plain-text, cipher-text, the four techniques for crypto-analysis, symmetric cryptography, asymmetric cryptography, digital signature, message authentication code, hash functions, and modes of encryption operations.

iii). Explain the concepts of malicious code, including virus, Trojan horse, and worms.

iv). Explain common vulnerabilities in computer programs, including buffer overflow vulnerabilities, time-of-check to time-of-use flaws, incomplete mediation.

v). Outline the requirements and mechanisms for identification and authentication.

vi). Explain issues about password authentication, including dictionary attacks (password guessing attacks), password management policies, and one-time password mechanisms.

Course learning Outcome

Upon the completion of the course, the student should be able to:

1. Describe the functioning of various types of malicious code, such as viruses, worms, trapdoors.
2. Enumerate a set programming techniques that enhances security.
3. Explain the various controls available for protection against internet attacks, including authentication, integrity check, firewalls, and intruder detection systems.
4. Describe the different ways of providing authentication of a user or program.
5. Describe the mechanisms used to provide security in programs, operating systems, databases and networks.
6. Describe the background, history and properties of widely-used encryption algorithms such as DES, AES, and RSA

Topics:

Basic Computer Security Concepts: **4 hours**

Threats, vulnerabilities, controls; risk; confidentiality, integrity, availability; security policies, security mechanisms; assurance; prevention, detection, deterrence.

Basic Cryptography: **5 hours**

Basic cryptographic terms; historical background; symmetric crypto primitives; modes of operation; cryptographic hash functions; asymmetric crypto primitives.

Program Security: **5 hours**

Flaws (malicious code - viruses, Trojan horses, worms; program flaws - buffer overflows, time-of-check to time-of-use flaws, incomplete mediation); defences (software development controls, testing techniques).

Security in Conventional Operating Systems: **6 hours**

Memory, time, file, object protection requirements and techniques; protection in contemporary operating systems; identification and authentication - identification goals, authentication requirements, human authentication, machine authentication.

Trusted Operating Systems: **4 hours**

Assurance; trust; design principles; evaluation criteria; evaluation process.

Database Management Systems Security: **6 hours**

Database integrity; database secrecy; inference control; multilevel databases.

Network Security: **6 hours**

Network threats (eavesdropping, spooling, modification, denial of service attacks); introduction to network security techniques (firewalls, virtual private networks, intrusion detection).

Management of Security: **5 hours**

Security policies; risk analysis; physical threats and controls.

Ethics, Social and Professional Issues: **4 hours**

Legal aspects of security; privacy and ethics.

Semester Total Hours **45 hours**

Study Material

Laptops with a modern operating system for which the student has administrator privileges, LCD projector, networking tools, printers and fast Internet.

Method of Delivery:

This Course will be delivered primarily through the lecture method complemented by students' group discussions. In addition, case studies will be used to enhance the students' analytical and communication skills.

Method of Assessment:

Assessment will be in terms of tests and practical exercises (30%) and annual examination (70%)

Recommended Reading/ Indicative Sources:

1. Michael Goodrich and Roberto Tamassia (2010). Introduction to Computer Security. 1st ed., Addison Wesley.
2. Charles P. Peeger and Shari L. Peeger (2003). Security in Computing. 3rd Edition. Prentice Hall. ISBN: 0-13-035548-8.
3. Matt Bishop (2004). Introduction to Computer Security. Addison-Wesley Professional. ISBN-10: 0321247442, ISBN-13: 978-0321247445.
4. https://www.cs.utexas.edu/users/byoung/cs361/syllabus361.html

Course Name:	**Modelling and Simulation**
Course Code:	**GCS3105**
Year of Study:	**3**
Semester:	**1**
Contact Hours:	**45**
Credit Units:	**3**

Course Description:

The course will introduce the basic concepts of computation through modeling and simulation that are increasingly being used by architects, planners, and engineers to shorten design cycles, innovate new products, and evaluate designs and simulate the impacts of alternative approaches. Students will use MATLAB to explore a range of programming and modeling concepts while acquiring those skills. They will then undertake a final project that analyzes one of a variety of scientific problems by designing a representative model, implementing the model, completing a verification and validation process of the model, reporting on the model in oral and written form, and changing the model to reflect corrections, improvements and enhancements.

Prerequisites: GCS 2102 Data Structures, Algorithms and Analysis.

Course Objectives:

1. Explain the benefits of simulation and modeling in a range of important application areas.

2. Demonstrate the ability to apply the techniques of modeling and simulation to a range of problem areas.
3. Evaluate a simulation, highlighting the benefits and the drawbacks.
4. Analyze modeling and simulation - Identify different types of models and simulations, describe the iterative development process of a model, and explain the use of models and simulations for hypothesis testing and explain how models link the physical world, the virtual world and the science of prediction.
5. Assess computational models - Discuss methods for reviewing models, their verification and validation. Discuss the differences between the predictions of the model, the actual results and the relevance of these differences to the problem. Discuss the suitability and limits of the model to address the problem for which the model was designed.
6. Complete a capstone modeling project that identifies a problem, develops a mathematical representation and transforms it to a computational model. Document the development and implementation of the model and present in oral and written form.

Learning Outcomes:

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

1. Demonstrate basic programming skills – functions, arrays, loops, conditional statements, procedures
2. Explain the Role of Modeling: Discuss the importance of modeling to science and engineering, the history and need for modeling, the cost effectiveness of modeling, the time-effect of modeling, define the terms associated with modeling to science and engineering, list questions that would check/validate model results, describe future trends and issues in science and engineering, and identify specific industry related examples of modeling in science and engineering.
3. Utilize the Modeling Process to identify the key parameters of a model, estimate model outcomes, utilize a computational tool, e.g., Matlab to implement the mathematical representation of the model, convey the results of the simulation accurately, validate the model with data, and discuss the quality and sources of errors in the model.
4. Explain and conduct the transforming of continuous functions and dynamics equations into discrete computer representations. Write pseudo-code for finite difference modeling equations and create a simulation in a computational tool, e.g., Matlab.
5. Examine mathematical representations of functions - Describe and utilize linear and nonlinear functions to model empirical data. Visualize empirical data and the fitting function using a computational tool.
6. Utilize Matlab as a computational tool - Describe the system syntax, define elementary representations, functions, etc. Explain programming and the scripting process, e.g., relational operations, logical representations, condition statements, loops, etc. Create tabular and graphical results.

Topics:

What is Simulation and Modeling? 2 hours

Definition of simulation and modeling; relationship between simulation and modeling.

Benefits and Limitations 4 hours

Role (addressing performance, optimization); supporting decision making, forecasting, safety considerations; for training and education.

Important Application Areas **4 hours**

Healthcare (including assisting with diagnostics); economics and finance; classroom of the future; training and education; city and urban simulations; simulation in science and in engineering; games; military simulation.

Types of Simulations **3 hours**

Physical; human in the loop; interaction; computer; virtual reality.

Simulation Process **3 hours**

Sound basis, identification of key characteristics or behaviours, simplifying assumptions; validation of outcomes.

Model Building **4 hours**

Using mathematical formula or equation, graphs, constraints.

Methodologies and Techniques **3 hours**

Use of Time stepping for Dynamic Systems **4 hours**

Theoretical Considerations **5 hours**

Monte Carlo methods; stochastic processes; queuing theory.

Simulation and Modeling Technologies **5 hours**

Graphics processors; haptic feedback devices; human computer interaction considerations.

Assessing and Evaluating Simulations **5 hours**

Simulation and Modeling Software **5 hours**

Packages; languages.

Semester Total Hours **45 hours**

Study Material

Laptops with a modern operating system for which the student has administrator privileges, LCD projector, networking tools, printers and fast Internet, MATLAB software.

Method of Delivery:

The course will be taught by using lectures, practical exercise, laboratory work, projects and assignments.

Method of Assessment:

Assignments, tests, exercises and final examination.

Their relative contributions to the final grade are: course work (assignments, tests, laboratory, and projects) 30%, final examination 70%, total 100%.

Reading Lists:

1. Bernard P. Zeigler, Herbert Praehofer and Tag Gon Kim (2000). Theory of Modeling and Simulation. 2nd ed., Academic Press.

2. John A. Sokolowski and Catherine M. Banks (2009). Principles of Modeling and Simulation: A Multidisciplinary Approach. Wiley.
3. Jerry Banks, John S. Carson, Barry L. Nelson and David M. Nicol (2009). Discrete-Event System Simulation. 5th ed., Prentice Hall.
4. Fishwick P. (1995). Simulation Model Design and Execution, PrenticeHall. ISBN 0-13-098609-7
5. Law A., Kelton D. (1991). Simulation Modelling and Analysis, McGraw-Hill. ISBN 0-07-100803-9
6. Rábová Z. a kol (1992): Modelování a simulace, VUT Brno. ISBN 80-214-0480-9
7. Ross, S. (2002). Simulation, Academic Press. ISBN 0-12-598053-1
8. Fishwick P. (1995): Simulation Model Design and Execution, PrenticeHall. ISBN 0-13-098609-7
9. Law A., Kelton D. (1991). Simulation Modelling and Analysis, McGraw-Hill; ISBN 0-07-100803-9
10. Texts available on WWW.

Course Name:	**Mobile Phone Application Development in Java**
Course Code:	**GCS 3106**
Year of Study:	**3**
Semester:	**1**
Contact Hours:	**45**
Credit Units:	**3**

Course Description:
Today's applications are increasingly mobile. Computers are no longer confined to desks and laps but instead live in our pockets and hands. This course teaches students how to build mobile apps for Android, iOS, and Windows 8, the trinity that is today's mobile operating platforms. Students learn to write both web apps and native apps for Android using Eclipse and the Android SDK, to write native apps for iPhones, iPod Touches, and iPads using Xcode and the iOS SDK, and to write web apps for both platforms. The course also touches on Windows 8 application programming, so as to provide students with a stepping stone for application development in the mobile operating system of their choice. Additional topics covered include application deployment and availability on the corresponding app stores and markets, application security, efficient power management, and mobile device security.

Prerequisites:

- Students are expected to have knowledge of HTML, Javascript and CSS. The course does not aim to cover these topics, though does discuss in detail the details and feature-sets of HTML5, and CSS3, as they pertain to mobile browsers. The first part of the course will deal with these subjects.
- Students are expected to be comfortable reading Java and/or C-code
- Students need to familiarize themselves with the development environments of Eclipse (free download), XCode and Visual Studio 2012 before the course starts.

Course Objectives:

1. Introduce the Java 2 Micro Edition (J2ME) and J2ME programming.
2. Describe programming details of the MIDP, including how to program the phone interface, handle events, make network connections, and work with databases.
3. Develop a MIDlet and subsequently convert it into an executable mobile phone application for devices supporting MIDP2.0 or higher.

Learning Outcomes:

On completion of this course, the students will be able to:

1. Understand the Java 2 Micro Edition (J2ME) and J2ME programming and latest technologies.
2. Describe programming details of the MIDP, program the phone interface, handle events, make network connections, and work with databases.
3. Develop a MIDlet and other new formats and convert them into an executable mobile phone application for devices supporting MIDP2.0 or higher.

Topics:

Instruction **2 Hours**

Introduction to Mobile Phone applications.

Revision on java (Understanding Object Oriented Programming) **9 Hours**

Basic Concepts; Conditions and Loops; Arrays; Classes and Objects; More on Classes; Exceptions, Lists, Threads & Files.

Introduction to android development **10 Hours**

Introduction to android Platform and architecture; Android SDK; Android User Interface; Resources, Views, and Intents; Intents and Storage; Storage (Using JSON) and Threads; Android UX.; Deployment.

J2ME mobility development **12 Hours**

J2ME Overview; Developing J2ME applications; J2ME configurations; J2ME profiles; Setting up development environment; CLDC API; Development using Kjava GUI components; Development using Kjava event handling; MIDP API; CDC API.

Introduction to USSD and SMS application development **10 Hours**

Overview of USSD; Understanding USSD API and the underlying concepts; Using Java to handle USSD; Example of USSD App; Overview of SMS technology; Under.

Conclusion and what's next **2 Hours**

Semester Total Hours **45 hours**

Study Material

Laptops with a modern operating system for which the student has administrator privileges, LCD projector, JDK, Android Studio, Visual Studio, printers and fast Internet.

Method of Delivery:

The course will be taught by using lectures, practical exercise, laboratory work, projects and assignments.

Method of Assessment:
Assignments, tests, exercises and final examination. Their relative contributions to the final grade are: course work (assignments, tests, laboratory, and projects) 30%, final examination 70%, total 100%.

Reading Lists
1. Kim Topley (2002). J2ME in a Nutshell. 1st ed., O'Reilly Media.
2. Qusay Mahmoud (2001). Learning Wireless Java. O'Reilly Media.
3. Jonathan Knudsen and Sing Li (2005). Beginning J2ME: From Novice to Professional. 3rd ed., Apress.

Course Name:	**Geographical Information Systems**
Course Code:	**GIT 3105**
Year of Study:	**3**
Semester:	**1**
Contact Hours:	**45**
Credit Units:	**3**

Course Description:
The course introduces principles, concepts and techniques of GIS. Students will also be introduced to theoretical, practical and application oriented aspects of GIS. The course covers the following: Spatial Data Management, Mapping with Google API, Creating Polylines and Polygons and ends with Geocoding web services using CSS and Java Script.

Course Objectives:
The course will support students to:
i) Build knowledge and understanding of GIS concepts and mapping theory.
ii) Develop skills in Geo-Spatial modelling.
iii) Examine how GIS is utilized in the larger context of business needs and IT strategies.
iv) Develop GIS prototypes using Geocoding techniques.

Course Learning Outcomes:
By the end of the course, the student should be able to:
i) Demonstrate understanding of GIS concepts and mapping theory.
ii) Perform Geo-Spatial modelling.
iii) Demonstrate understanding of the basic concepts of geography necessary to efficiently and accurately use GIS technology and data concepts.
iv) Apply Geocoding techniques in developing relevant prototypes.

Detailed Course Content

Introduction to GIS and Remote Sensing	**3 hours**
Spatial Data Management	**9 hours**

Mapping with Google API	**9 hours**
Creating Polylines and Polygons	**6 hours**
Location and coordinate system	**6 hours**
Geocoding web services using CSS and Java Script	**12 hours**
Semester Total	**45 hours**

Study Materials
Desktops/laptops with a modern operating system for which relevant scripting IDEs are installed, ArcGIS, ArcMap, GPS Device, ArcInfo, LCD projector and fast Internet

Mode of Delivery
Lectures, lab-sessions and instructor-led group discussions on real life mapping case studies will be the main method of instruction.

Mode of Assessment
Assessment will be in form of assignments, tests, lab exercises and final examination. Their relative contributions to the final grade are: course work (assignments, tests, laboratory work) 30%, final examination 70%, total 100%.

Reference List

1. Njoku, E.G. (Ed.) (2014). Encyclopedia of Remote Sensing. New York, NY: Springer Science.
2. Svennerberg, G. (2010). Beginning Google Maps API 3. New York, NY: Springer Science
3. John R. Jensen, (2000). Remote Sensing of the Environment, Prentice Hall

Course Name:	**Advanced Database Systems**
Course Code:	**GCS 3201**
Year of Study:	**3**
Semester:	**2**
Contact Hours:	**45**
Credit Units:	**3**

Description
This course introduces the theoretical and practical issues relating to the design and use of database systems. In addition to the provision of a very sound foundation in traditional, second generation database systems it explores the representation and management of complex information resources with third generation database technology. The course highlights how database systems are at the core of modern information-rich Websites, and explores the role of XML and Semantic Web technologies in creating a “Web of data".

Prerequisites: GCS 2106 Databases.

Course Objectives

This course aims to:

i). Provide an understanding the latest types of DBMS - object oriented, extended relational, XML.

i). Introduce the context in which XML may be used in database systems.

ii). Explain the role of database systems in information-rich Websites.

iii). Discuss how XML and Semantic Web technologies are used to model and manage information in a Web context.

iv). Compare alternative DBMS schemas for a conceptual model assessing their strengths and weaknesses.

Course Learning Outcome

On completion of this course, the students will be able to:

1. Link databases to programs and web interfaces.
2. Create XML and Semantic Web information models.
3. Analyse a conceptual schema and transform it into the schema of an appropriate database management system.
4. Design object oriented and extended relational schemas.
5. Analyse a Website from an information modelling perspective.

Topics

Overview: **3 hours**

Database environment; database architectures and the Web.

Selected Database Issues: **6 hours**

Security and administration; professional, legal, and ethical issues; transaction management; query processing.

Distributed DBMSs and Replication: **6 hours**

Distributed DBMSs concepts and design; distributed DBMSs advanced concepts; replication and mobile databases.

Object-Oriented DBMSs: **10 hours**

Object-Oriented DBMSs concepts and design; Object-Oriented DBMSs standards and languages; Object-Relational DBMSs.

Web and DBMSs: **8 hours**

Web technology and DBMSs; semi structured data and XML.

Business Intelligence Technologies: **12 hours**

Data warehousing concepts; data warehousing design; OLAP; data mining.

Semester Total Hours **45 hours**

Study Material

Laptops with a modern operating system for which the student has administrator privileges, LCD projector, Oracle server, SQL Server, Microsoft Visio, printers and fast Internet.

Method of Delivery:
The course will be taught by using lectures, practical exercise, laboratory work, projects and assignments.

Method of Assessment:
Assignments, tests, exercises and final examination. Their relative contributions to the final grade are: course work (assignments, tests, laboratory, and projects) 30%, final examination 70%, total 100%.

Reading Lists

1. Thomas M. Connolly and Carolyn E. Begg (2015). Database Systems: A Practical Approach to Design, Implementation and Management. 6th ed., Pearson.
2. Suzanne W Dietrich and Susan D. Urban (2004). An Advanced Course in Database Systems: Beyond Relational Databases. Prentice Hall.
3. Hector Garcia-Molina, Jeffrey D. Ullman, and Jennifer Widom (2008). Database Systems: The Complete Book. 2nd ed., Prentice Hall.

Course Name:	**Parallel Computation**
Course Code:	**GCS 3202**
Year of Study:	**3**
Semester:	**2**
Contact Hours:	**45**
Credit Units:	**3**

Description
The course surveys parallel computation across hardware, software, programming languages, algorithms and applications areas. The emphasis is on learning enough context in each topic area to support effective parallel programming.

Prerequisites: GCS 2104 Computer Architecture, GCS 2202 Operating Systems,
GCS 3103 Algorithm Design and Analysis.

Course Objectives
The course objectives are to provide students with:

1. Compare and contrast architectures for parallel computing, recognizing the strengths and weaknesses of each.
2. Compare and contrast parallel programming paradigms recognizing the strengths and weaknesses of each.
4. Identify the basic properties of bandwidth, latency, scalability and granularity
5. Design, code, test and debug programs for parallel computation

Course Learning Outcome
On completion of this course, the students will be able to:

1. Explain how large-scale parallel systems are architectured and how massive parallelism are implemented in accelerator architectures;

2. Write parallel programs for large-scale parallel systems, shared address space platforms, and heterogeneous platforms;
3. Design efficient parallel algorithms and applications;
4. Be conversant with performance analyze and modeling of parallel programs;

Topics

Context for Today's Parallel Computing: 4 hours

Everyday parallelism, parallelism in processor architecture, parallel computer structures from multicore to huge clusters, etc.

Basic Concepts: 5 hours

Models of parallel computers, differences (and similarities) between parallel and distributed computation, data and task parallelism, threads and processes, latency and bandwidth, locality, concurrent and exclusive reads and writes, dependences, MIMD/SIMD, Amdahl's Law, performance measurements, true speed-up, relative speed-up, efficiency, super-linear speed-up.

Shared Memory Parallel Machine Architecture Concepts: 3 hours

Multi-threading, multi-core, SMP, snooping and directory-based coherency protocols, sequential consistency, private and shared memory.

Distributed Memory Parallel Machine Architecture Concepts: 3 hours

Disjoint address spaces, globalization of address spaces, message passing, synchronous and asynchronous communication.

Interconnection Networks: 3 hours

Common topologies, bisection bandwidth, bus and cross-bar connectivity.

Basics of Threaded Parallel Computation: 3 hours

Problem partitioning, hazards of concurrency, locking, lock contention, thrashing, privatization, false sharing, serialization.

Algorithmic Issues: 4 hours

Problem decomposition, memory reference costs, techniques for improving locality, parallel prefix, controlling granularity and dependences, block and cyclic storage organization, tree decompositions, static and dynamic task assignment.

Languages and Libraries: 5 hours

Languages and libraries for threaded parallel programming, POSIX threads, Java threads and memory model, automatic threading systems (OpenMP); Languages and libraries for distributed memory parallel programming, PVM and MPI, partitioned global address space languages (PGAS).

Higher Level Approaches: 3 hours

Array-based, functional, domain-specific.

Transaction Approach: 3 hours

Transaction approach to memory consistency, hardware techniques, software techniques, rollback.

Emerging Languages: 3 hours

Emerging languages including high-productivity parallel languages.

Co-processor Techniques: 3 hours

Co-processor techniques including GPU, Cell, FPGA, characteristics of co-processor programming methodologies.

Experimental Techniques: 3 hours

Experimental techniques, measuring performance, computing speedup, experimental precautions, controlling variation, experimental dangers, reporting performance.

Semester Total Hours 45 hours

Study Material

Laptops with a modern operating system for which the student has administrator privileges, LCD projector, networking tools, printers and fast Internet.

Method of Delivery:

The course will be taught by using lectures, practical exercise, laboratory work, projects and assignments.

Method of Assessment:

Assignments, tests, exercises and final examination. Their relative contributions to the final grade are: course work (assignments, tests, laboratory, and projects) 30%, final examination 70%, total 100%.

Reading Lists

1. Ananth Grama, George Karypis, Vipin Kumar and Anshul Gupta (2003). Introduction to Parallel Computing. 2nd ed.
2. Maurice Herlihy and Nir Shavit (2008). The Art of Multiprocessor Programming. Morgan Kaufmann.
3. Jack Dongarra, Ian Foster, Geo_rey C. Fox, William Gropp, Ken Kennedy, Linda Torczon and Andy White (2002). The Sourcebook of Parallel Computing. Morgan Kaufmann.

Course Name:	**Logic Programming**
Course Code:	**GCS3203**
Year of Study:	**3**
Semester:	**2**
Contact Hours:	**45**
Credit Units:	**3**

Description

Logic programming is a declarative programming paradigm based on mathematical logic that is well-suited to build intelligent systems. This course introduces students to Logic Programming, with particular emphasis on the common foundations: declarative representation and deduction.

Prerequisites: GCS 2102 Data Structures, Algorithms and Analysis.

Course Objectives

The course objectives are to provide students with:

1. Understand the main principles of logic programming and will have learnt how to apply them in various logic programming environments.
2. Learnt programming skills using the Prolog language, with an emphasis on recursion and list processing.
3. Got to grips with the basics of first-order logic including: syntax, semantics, entailment, proof (soundness, completeness) and its use in formalising a domain.

Course Learning Outcome

By the end of the subject, students should:

1. be conversant with the syntax and semantics of propositional and predicate logic
2. be familiar with a variety of applications of predicate logic in software verification, databases and knowledge-based systems
3. be able to write specifications in predicate logic expressing state constraints
4. understand the notion of formal proof, and be able to construct simple proofs in a natural deduction proof system for predicate logic
5. be aware that there are inherent expressiveness and computational limitations to the applicability of logical systems, and be familiar with a number of restrictions under which the computational limitations can be overcome
6. be able to write programs in a logic programming language,
7. understand both the top-down and the bottom up operational semantics of logic programs
8. be familiar with a logic for reasoning about sequential programs, and capable of constructing correctness proofs for simple programs

Topics

Logic Programming: **10 hours**

Horn clauses, substitutions and unifiers, unification algorithm.

Operational Semantics: **10 hours**

Resolution, SLD resolution, SLD derivations, SLD trees.

Declarative Semantics: **10 hours**

Least Herbrand model, fix point construction, correctness and completeness.

Programming in Prolog: **11 hours**

Lists; arithmetic; trees; accumulation technique – difference lists; Cut.

Advanced Prolog Programming: **10 hours**

Meta-programs, non-determinism, constraints.

Program Analysis and Verification: **9 hours**

Termination, well-moding, well-typing.

Semester Total **45 hours**

Study Material
Laptops with a modern operating system for which the student has administrator privileges, LCD projector, networking tools, printers and fast Internet.

Method of Delivery:
The course will be taught by using lectures, practical exercise, laboratory work, projects and assignments.

Method of Assessment:
Assignments, tests, exercises and final examination. Their relative contributions to the final grade are: course work (assignments, tests, laboratory, and projects) 30%, final examination 70%, total 100%.

Reading Lists

1. Stuart Russell and Peter Norvig (2008). Artificial Intelligence: A Modern Approach. Prentice-Hall of India. ISBN: 8120323823.
2. Christopher John Hogger (1984). Introduction to Logic Programming. Academic Press. ISBN: 0123520908.
3. J. M. Spivey (1996). An Introduction To Logic Programming Through Prolog. Prentice Hall. ISBN: 0135360471.
4. Peter Flach (1998). Simply Logical - Intelligent Reasoning by Example. John Wiley. ISBN: 0471-94152-2.

Course Name:	**Capstone CS Project**
Course Code:	**GCS 3204**
Year of Study:	**3**
Semester:	**2**
Contact Hours:	**60**
Credit Units:	**4**

Description:
The senior capstone CS project covers project proposal writing, feasibility studies, intellectual property, teamwork, budgets, schedule management; professional communications (reports and presentations), design, implementation, testing. The course gives students the opportunity to show individual creativity and originality, to apply where appropriate knowledge and skills taught throughout CS programme, and to demonstrate investigative, problem-solving and other transferable skills.

Prerequisites: All courses in the previous Semesters relevant for successful completion.

Course Objectives

i. Summarize existing and background work relevant to the project.
ii. Use appropriate experimental techniques and validation methods in developing solutions.
iii. Critically evaluate solutions and findings resulting from the project.
iv. Plan, organize and implement tasks within time constraints.
v. Work independently under the project supervisor's direction.
vi. Report, present and document the findings and deliverables resulting from the project.

Course Learning Outcome:

i. Undertake a non-trivial computing science project.
ii. Understand the elements of a successful computing science project.
iii. Show an appreciation of best practice in solving computing science problems.
iv. Demonstrate an in-depth understanding of the technology and methodology used in the project.
v. Apply appropriate theory and technology to solving specific computing science problems.
vi. Develop computing science solutions rationally using a disciplined approach.

Topics:
Cornerstone Project Coverage: **60 hours**
Background study to become aware of current developments in the area of activity related to the project. Development of some tangible piece of software, hardware, system design or theoretical result. This need not necessarily be a usable finished project. Instead it could be, for example, an extension to an existing system, or a prototype built as part of a feasibility study. Deliverables do not necessarily have to be programs. They could be in non-executable form, for example, an SSM conceptual model. Summary of project findings and achievement in the form of a written report, up to 15,000 words in length.

Semester Total Hours **60 hours**

Study Material
Laptops with a modern operating system for which the student has administrator privileges, LCD projector, networking tools, printers and fast Internet.

Method of Delivery:
Students agree a project with a staff member (who becomes the supervisor) towards the end of the second year. Each student is supervised by the supervisor and is moderated by a moderator (another staff member) for the duration of the project. The student is required to submit, at regular intervals, progress reports/deliverables to the supervisor.
The student gives an interim presentation, submits the final project report and attends a project viva at the end of the project (towards the end of the semester). Students are expected to undertake about 300 hours of study for this course.

Method of Assessment:
Interim presentations, final project report and project viva. Their relative contributions to the final grade are: course work (interim presentations) 30%, final project reports and project viva 70%, total 100%.

Reading Lists:

1. Christian Dawson (2009). Projects in Computing and Information Systems: A Student's Guide. 2nd ed., Addison Wesley.
2. Christian Dawson (2000). The Essence of Computing Projects, A Student's Guide. 1st ed., Prentice Hall.
3. Ian Ricketts (1998). Managing Your Software Project: A Student's Guide. Springer.

Course Name:	**Mobile Web Application Development**
Course Code:	**GCS 3205**
Year of Study:	**3**
Semester:	**2**
Contact Hours:	**45**
Credit Units:	**3**

Description:

This course equips students with the skills to design mobile phone applications specifically for the needs of people in the developing world. While mobile phones are becoming dominant computing platform in developing world, most computer science courses in developing world currently focus exclusively on programming traditional desktop computers. The entrepreneurial emphasis of this courses will lead to dozens of applications designed specifically for markets in developing world.

Prerequisites: GCS 1202 Introduction to Object Oriented Programming, GCS 1204 Introduction to Internet Programming, GIT 3103 IT Project Management.

Course Objectives

The purpose of this course is to:

1. Understand the social impact that mobile technologies are having in the life of low-income people in developing countries, and to chart their possibilities for the future.
2. Design and launch mobile technologies that are technically appropriate and socially informed in the context of developing countries, so as to enable true and sustainable adoption for the next billion users.
4. Learn to overcome the non-technical barriers (social, educational, industrial, and financial) that prevent social mobile technologies from large scale deployment in commercial networks.
5. Help shape the vision of how pervasive connectivity can create unprecedented opportunities for empowering low-income people in developing countries.

Course Learning Outcomes

Upon successful completion of the course, the student will demonstrate the ability to:

1. Explain mobile devices, including their capabilities and limitations.
2. Use current mobile platforms and their architectures.
3. Develop mobile applications on a popular mobile platform.
4. Evaluate development with another mobile platform.

Topics

Introduction **2 Hours**

Mobile Developer Tools **2 Hours**

Dev tools for Desktop; Mobile Dev tools; Set up for mobile; Using Dev tools on mobile; Reverse port forwarding; Mobile tools for iOS.

Mobile UX and Viewport **3 Hours**

Concept of "Mobile First" design and development; Constraints of the canvas; Screen size for mobile; Fixed width; Rescaling on devices; Changing viewport settings; Fixed vs. device widths; Separate Desktop and Mobile; Rules for Mobile Only; Building Fluid Designs.

Fluid Design **3 Hours**

Making Fluid layouts with different frameworks (such as Bootstrap, etc.); Flexbox and its compatibility; Fluid to responsive design.

Media Queries **5 Hours**

Different devices different designs; Reflow across all devices; changing layouts; Media quires; Mobile device pixels and ratio.

Responsive Images **5 Hours**

Beautiful/High quality images vs. low quality; Responsive images; Ideal image solution.

Optimizing Performance **4 Hours**

Importance of performance; Analyzing performance; Compute performance; Profiling performance; Network aspects; Battery drain; Network usage pattern analysis; Minimizing full power radio; Network latency; Memory constraint problems; Handling garbage collection; Animation API; Optimizing animation for battery usage.

Touch **4 Hours**

Touch interaction; Simple rules for touch UI; Supporting touch and Mouse on UI; Touch events and libraries; Gesture libraries.

Input **5 Hours**

Input keyboards; Input attributes; Regex validation; Wrapping in labels; Input validation issues; Telephone calls.

Device Access **6 Hours**

Sensors on mobile; Camera access and capture extension; Audio input; Geolocation API; Device orientation and motion; Possibilities of device access.

Offline and Storage **5 Hours**

Network as an enhancement; "Offline first" development; Appcache and HTTP cache; Local data storage; WebSQL and IndexedDB.

Conclusion and Next Steps **2 Hours**

Deploying to app stores.

Semester Total Hours **45 hours**

Study Material
Laptops with a modern operating system for which the student has administrator privileges, LCD projector, networking tools, printers and fast Internet.

Method of Delivery:
This Course will be delivered primarily through the lecture method complemented by students' group discussions. In addition, case studies will be used to enhance the students' analytical and communication skills.

Method of Assessment:
Assessment will be in terms of tests and practical exercises (30%) and annual examination (70%)

Reading Lists

1. Jurgen Scheible and Ville Tuulos (2007). Mobile Python: Rapid Prototyping of Applications on the Mobile Platform. Wiley.
2. Li, S., and J. Knudsen (2005). Beginning J2ME: From Novice to Professional. 3rd ed. Berkeley, CA: Apress. ISBN: 9781590594797.
3. Allen Downey (2008). Think Python: How to Think Like a Computer Scientist. Green Tea Press.
5. Allen Downey (2009). Python for Software Design: How to Think Like a Computer Scientist. Cambridge University Press.
6. Kim Topley (2002). J2ME in a Nutshell. 1st ed., O'Reilly Media.
7. Qusay Mahmoud (2001). Learning Wireless Java. O'Reilly Media.

Course Name:	**Information Retrieval and Web Search**
Course Code:	**GCS 3206**
Year of Study:	**3**
Semester:	**2**
Contact Hours:	**45**
Credit Units:	**3**

Description:
The course covers traditional material as well as recent advances in information retrieval (IR), the study of the processing, indexing, querying, organization, and classification of textual documents, including hypertext documents available on the WWW.

Prerequisites: GCS 1202 Introduction to Object Oriented Programming, GCS 2102 Data Structures, Algorithms and Analysis.

Course Objective
The main objectives of this course are:

1. Understand the basic information storage and retrieval concepts.
2. Describe what issues are specific to efficient information retrieval.

3. Give applications of alternative search strategies and explain why the particular search strategy is appropriate for the application.
4. Perform Internet-based research.
5. Design and implement a small to medium size information storage and retrieval system.

Course Learning Outcome

At semester’s end, students will be able to:

1. Explain basic information storage and retrieval concepts.
2. Explain issues that are specific to efficient information retrieval.
3. Describe applications of alternative search strategies and explain why the particular search strategy is appropriate for the application.
4. Search for information on the Internet.
5. Develop small to medium size information storage and retrieval systems.

Topics

Overview of Information Retrieval: 3 hours

Goals and history of Information Retrieval (IR); impact of the web on IR.

Basic IR Models: 5 hours

Boolean and vector-space retrieval models; ranked retrieval; text similarity metrics; TF-IDF (term frequency/inverse document frequency) weighting; cosine similarity.

Basic Tokenizing, Indexing, and Vector Space Retrieval: 4 hours

Simple tokenizing, stop-word removal, and stemming; inverted indices; efficient processing with sparse vectors; Java implementation.

Experimental Evaluation of IR: 3 hours

Performance metrics: recall, precision, and F-measure; evaluations on benchmark text collections.

Query Operations and Languages: 5 hours

Relevance feedback; query expansion; query languages.

Text Representation: 3 hours

Word statistics; Zipf's law; Porter stemmer; morphology; index term selection; using thesauri. Metadata and markup languages (SGML, HTML, XML).

Web Search: 4 hours

Search engines; spidering; metacrawlers; directed spidering; link analysis (e.g. hubs and authorities, Google PageRank); shopping agents.

Text Categorization: 4 hours

Categorization algorithms: Rocchio, nearest neighbour, and naïve Bayes. Applications to information filtering and organization.

Language Model Based Retrieval: 4 hours

Using naive Bayes text classification for ad hoc retrieval. Improved smoothing for document retrieval.

Text Clustering: **4 hours**

Clustering algorithms: agglomerative clustering; k-means; expectation maximization (EM). Applications to web search and information organization.

Recommender Systems: **3 hours**

Collaborative filtering and content-based recommendation of documents and products.

Information Extraction and Integration: **3 hours**

Extracting data from text; semantic web; collecting and integrating specialized information on the web.

Semester Total Hours **45 hours**

Study Material

Laptops with a modern operating system for which the student has administrator privileges, LCD projector, networking tools, printers and fast Internet.

Method of Delivery:

This Course will be delivered primarily through the lecture method complemented by students' group discussions. In addition, case studies will be used to enhance the students' analytical and communication skills.

Method of Assessment:

Assessment will be in terms of tests and practical exercises (30%) and annual examination (70%)

Textbooks Required:

1. Christopher D. Manning, Prabhakar Raghavan and Hinrich Schutze (2008). Introduction to Information Retrieval. Cambridge University Press.
2. Bruce Croft, Donald Metzler and Trevor Strohman (2009). Search Engines: Information Retrieval in Practice. Pearson Education.
3. Ricardo Baeza-Yates and Berthier Ribeiro-Neto (2011). Modern Information Retrieval. 2nd ed., Addison-Wesley Professional.
4. Richard K. Belew (2001). Finding Out About. Cambridge University Press.
5. Karen Sparck Jones and Peter Willett (1997). Readings in Information Retrieval. Morgan Kaufmann Publishers, Inc.
6. Soumen Chakrabarti (2003). Mining the Web: Discovering Knowledge from Hypertext Data. Morgan-Kaufmann Publishers.
7. Fah-Chun Cheong (19996). Internet Agents: Spiders, Wanderers, Brokers, and Bots. New Riders Pub.
8. Ian H. Witten, Alistair Mofat, and Timothy C. Bell (1999). Managing Gigabytes: Compressing and Indexing Documents and Images. 2nd ed., Morgan Kaufmann.
9. Haralambos Marmanis and Dmitry Babenko (2009). Algorithms of the Intelligent Web. 1st ed., Manning Publications.

Course Name:	**Linear Algebra**
Course Code:	**GCS 3207**
Year of Study:	**3**
Semester:	**2**
Contact Hours:	**45**
Credit Units:	**3**

Description:

This course covers matrix theory and linear algebra, emphasizing topics useful in other disciplines. Linear algebra is a branch of mathematics that studies systems of linear equations and the properties of matrices. The concepts of linear algebra are extremely useful in physics, economics and social sciences, natural sciences, and engineering. Due to its broad range of applications, linear algebra is one of the most widely taught subjects in college-level mathematics (and increasingly in high school).

Course objectives:

This course aims to:

1. Solve systems of linear equations and homogeneous systems of linear equations by Gaussian elimination and Gauss-Jordan elimination.
2. Row-reduce a matrix to either row-echelon or reduced row-echelon form.
3. Use matrix operations to solve systems of equations and be able to determine the nature of the solutions.
4. Understand some applications of systems of linear equations.
5. Perform operations with matrices and find the transpose and inverse of a matrix.
6. Calculate determinants using row operations, column operations and expansion down any column and across any row.
7. Interpret vectors in two and three-dimensional space both algebraically and geometrically.
8. Use the Gram-Schmidt process to produce an orthonormal basis.
9. Use basic mathematical proof techniques to prove or disprove certain claims (e.g. prove or disprove whether a given set of objects constitutes a vector space).
10. Find the kernel, range, rank, and nullity of a linear transformation.
11. Calculate eigenvalues and their corresponding Eigen spaces.
12. Understand the concept of a linear transformation as a mapping from one vector space to another and be able to calculate its matrix representation with respect to standard and nonstandard bases.
13. Determine if a matrix is diagonalizable, and if it is, how to diagonalizable it.

Course Learning Outcomes

At the end of this course the successful student will be:

1. Familiar with the ideas of matrices and their applications in solving problems involving systems of linear equations and linear programming problems.
2. Capable of representing geometric transformations by means of matrices and to express the volume of certain figures and equation of line using determinants.

Topics

Systems of Linear Equations 9 Hours

Introduction to Systems of Linear equations. Gaussian Elimination and Gauss Jordan Elimination. Application of Systems of Linear Equations.

Applications of Systems of equations and Matrices **6 Hours**

Application of Systems of Linear Equations Continued. Operations with Matrices. Properties of Matrix Operations. The Inverse of a Matrix. Elementary Matrices. Application of Matrix operation.

Determinants and their Applications **9 hours**

The Determinant of a Matrix. Evaluation of a Determinant using Elementary Operations. Properties of Determinants. Introduction of Eigenvalues. Applications of Determinants.

Vector Spaces **6 Hours**

Vectors in R^n. Vector Spaces. Subspaces of Vector Spaces. Spanning Sets and Linear Independence. Bases and Dimension. Rank of a Matrix and Systems of Linear Equations. Coordinates and change of Basis. Application of Vector Spaces.

Inner Product Space **6 Hours**

Length and Dot Product in R^n. Inner Product Space. Orthonormal Basis: Gram-Schmidt Process. Mathematical Models and Least Square Analysis. Application of Inner Product Space.

Linear Transformations **9 Hours**

Introduction to Linear Transformations. The Kernel and Range of a Linear Transformation. Matrices for Linear Transformations. Transition Matrices and Similarity. Application of Linear Transformations

Semester Total **45 hours**

Study Material

Laptops with a modern operating system for which the student has administrator privileges, LCD projector, whiteboards & markers, printers and fast Internet.

Method of Delivery:

This Course will be delivered primarily through the lecture method complemented by students' group discussions. In addition, case studies will be used to enhance the students' analytical and communication skills.

Method of Assessment:

Assessment will be in terms of tests and practical exercises

Reading Lists

1. Linear Algebra With applications, 4th ed., Prentice Hall, New Jersey 2002
2. Larson, R (2017). Elementary linear algebra. 8th edition. Boston, MA: Brooks/Cole, Cengage Learning. ISBN: 978-1-305-65800-4
3. Larson, R (2017). Student solutions manual for elementary linear algebra. 8th edition. Boston, MA: Brooks/Cole, Cengage Learning. ISBN: 978-1-305-65802-8

Course name: **Emerging Trends in Computer Science**

Course Code:	**GCS 3208**
Year of Study:	**3**
Semester:	**2**
Contact Hours:	**45**
Credit Units:	**3**

Description:

The field of computer science is experiencing a transition from computation-intensive to data-intensive problems, wherein data is produced in massive amounts by large sensor networks, new data acquisition techniques, simulations, and social networks. Efficiently extracting, interpreting, and learning from very large datasets requires a new generation of scalable algorithms as well as new data management technologies.

The course is to expose or provide students with an opportunity to search for knowledge in an emerging technological trend. It is to allow a student do lightweight research and explore the current trends in a certain computer science area.

Course Objective

The aims of the course are:

i). To aid a student get an in depth understanding of the developments in one area of computer science;

ii). To improve the student's research skills in line with emerging technologies;

iii). To develop confidence in the students on the ability to search for knowledge with little guidance;

iv). To provide an understanding of professional, ethical and social responsibilities;

v). To introduce the concepts of new Computing disciplines coming into the current IT market;

Course Learning Outcome

At semester's end, students will have:

i). An understanding of professional, ethical and social responsibilities;

ii). An ability to analyze the impact of computing on individuals, organizations, and society, including ethical, legal, security and global policy issues;

iii). Ability to recognize of the need for, and an ability to engage in, continuing professional development;

iv). An ability to apply knowledge of computing and mathematics appropriate to the discipline;

v). An ability to communicate orally with a range of audiences;

vi). Broad understanding of Big Data application areas and approaches used. Good preparation for any student likely to be involved with Big Data in their future;

vii). An understanding of Cloud Computing and its applications;

viii). Understanding of how computer technologies are being used to support education and other fields like Bioinformatics and cyber security.

Topics/Indicative content

Introduction to emerging trends **2 Hours**

Brief introduction to emerging trends.

Big Data Analytics **10 Hours**

Big Data - Beyond the Hype; what is Big Data; The Big Data Platform; Five High Value Big Data Use Cases; Technical Details of Big Data Components.

Cloud Computing 9 Hours
Cloud computing introduction; cloud computing for everyone; using cloud services; outside the cloud; storing and sharing.

Computer Assisted Education 9 Hours
Computer based training; game-based learning; computer assisted instruction; e-learning platforms;

Bioinformatics 9 Hours
Application of bioinformatics in Biology, Biomedical engineering and Medicine.

Cyber security 6 Hours
Social engineering, computer forensics, ethical hacking; Security architecture principles and frameworks (i.e. SABSA, Zachman, TOGAF, etc.); OSI model; TCP/IP; General firewall features, types, issues, and platforms; Networking (i.e. ports, protocols, VPNs, etc.).

Semester Total Hours 45 hours

Study Material:
Laptops with a modern operating system for which the student has administrator privileges, LCD projector, networking tools, printers and fast Internet.

Method of Delivery:
This Course will be delivered primarily through the lecture method complemented by students' group discussions. In addition, case studies will be used to enhance the students' analytical and communication skills.

Method of Assessment:
Assessment will be in terms of tests and practical exercises (30%) and annual examination (70%)

Reading Lists:

1. Allamaraju, S. (2011). Professional Java Server Programming J2EE (4.1 Edition). Wrox Press Ltd.
2. Beaulieu, M. (2012). Wireless Internet applications & Architectures: Building Professional Wireless Applications Worldwide, Addison-Wesley.
3. Burd, B. (2014). Java Programming for Android Developers for Dummies. Hoboken, NJ: John Wiley & Sons, Inc.
4. Erl, T., & Puttini, R. (2013). Cloud Computing: Concepts, Technology & Architecture. Kindle eBook: Prentice Hall.
5. Horton, J. (2015). Android Programming for Beginners. Birmingham, UK: Packet Publishing.
6. Marr, B. (2015). Big Data: Using SMART Big Data, Analytics and Metrics to Make Better Decisions and Improve Performance. Kindle eBook.

Appendix A: Programme Budget

Costing

The programme shall be run on cost-recovery basis. The rate shall from time to time be determined by the University Council. Programme enrolment per intake is 40 (forty) students private sponsored respectively. Table 1 shows expect income and expenditure for the programme

Expected Enrolment = 40

Year 1 Semester 1				
Income: 40 x 910,000 = 36,400,000				
S/N	**Item**	**Unit cost**	**Quantity**	**Total**
1	**Allowance**			
	Teaching allowance for 300 contact hours	32,000	300	9,600,000
	Support staff allowance	5,000	300	1,500,000
2	**Examination**			
	Examination Materials	1,500,000		1,500,000
	Invigilation of six exams including refreshments	9,000	6	54,000
	Marking allowance for 240 scripts per course for 5 courses	1,500	240	360,000
	External Examiners	1,000,000		1,000,000
3	**Machine Equipment Maintenance**			
	Servicing Computers	30,000	120	360,000
	Computers Repair and maintenance	20,000	120	240,000
	Printers and Photocopy Maintenance	50,000	4	200,000
	Antivirus Purchase	50,000	10 Pieces	500,000
4	**Teaching Materials**			
	Whiteboard Markers	10,000	10 Boxes	100,000
	Chalks	5,000	2 boxes	10,000
	Board dusters	5,000	2 Boxes	10,000
	Printing and Photocopying Papers	75,000	2 Cartons	150,000
	Writing Pens	10,000	3 Boxes	30,000
5	**Administrative Cost**			
	Repair of chairs and tables	10,000	45	450,000
	Repair of locks and window fasteners	10,000	15	150,000
	Electrical Maintenances	100,000	Units	100,000
	Generator Fuel	3,500	1000 Liters	3,500,000
	Electricity Bills	600	700 units	420,000
	Water Bills	2,500	200 units	500,000
Total				**20,734,000**
Year 1 Semester 2				
Income: 40 x 910,000 = 36,400,000				
1	**Allowance**			

	Teaching allowance for 300 contact hours	32,000	300	9,600,000
	Support staff allowance	5,000	300	1,500,000
2	**Examination**			
	Examination Materials	1,500,000		1,500,000
	Invigilation of six exams including refreshments	9,000	6	54,000
	Marking allowance for 240 scripts per course for 5 courses	1,500	240	360,000
	External Examiners	1,000,000		1,000,000
3	**Machine Equipment Maintenance**			
	Servicing Computers	30,000	120	360,000
	Computers Repair and maintenance	20,000	120	240,000
	Printers and Photocopy Maintenance	50,000	4	200,000
	Antivirus Purchase	50,000	10 Pieces	500,000
4	**Teaching Materials**			
	Whiteboard Markers	10,000	10 Boxes	100,000
	Chalks	5,000	2 boxes	10,000
	Board dusters	5,000	2 Boxes	10,000
	Printing and Photocopying Papers	75,000	2 Cartons	150,000
	Writing Pens	10,000	3 Boxes	30,000
5	**Administrative Cost**			
	Repair of chairs and tables	10,000	45	450,000
	Repair of locks and window fasteners	10,000	15	150,000
	Electrical Maintenances	100,000	Units	100,000
	Generator Fuel	3,500	1000 Liters	3,500,000
	Electricity Bills	600	700 units	420,000
	Water Bills	2,500	200 units	500,000
Total				**20,734,000**
Recess Semester				
1	Teaching allowance for 135 contact hours	32,000	135	4,320,000
2	Invigilation of two exams including refreshments	9,000	2	18,000
3	Marking allowance for 40 scripts per course for 2 courses	1,500	80	120,000
Total				**4,458,000**
Year 2 Semester 1				
Income: 40 x 910,000 = 36,400,000				
1	**Allowance**			
	Teaching allowance for 300 contact hours	32,000	300	9,600,000
	Support staff allowance	5,000	300	1,500,000
2	**Examination**			
	Examination Materials	1,500,000		1,500,000

	Invigilation of six exams including refreshments	9,000	6	54,000
	Marking allowance for 240 scripts per course for 5 courses	1,500	240	360,000
	External Examiners	1,000,000		1,000,000
3	**Machine Equipment Maintenance**			
	Servicing Computers	30,000	120	360,000
	Computers Repair and maintenance	20,000	120	240,000
	Printers and Photocopy Maintenance	50,000	4	200,000
	Antivirus Purchase	50,000	10 Pieces	500,000
4	**Teaching Materials**			
	Whiteboard Markers	10,000	10 Boxes	100,000
	Chalks	5,000	2 boxes	10,000
	Board dusters	5,000	2 Boxes	10,000
	Printing and Photocopying Papers	75,000	2 Cartons	150,000
	Writing Pens	10,000	3 Boxes	30,000
5	**Administrative Cost**			
	Repair of chairs and tables	10,000	45	450,000
	Repair of locks and window fasteners	10,000	15	150,000
	Electrical Maintenances	100,000	Units	100,000
	Generator Fuel	3,500	1000 Liters	3,500,000
	Electricity Bills	600	700 units	420,000
	Water Bills	2,500	200 units	500,000
Total				**20,734,000**
	Teaching allowance for 300 contact hours	32,000	300	9,600,000
	Support staff allowance	5,000	300	1,500,000
2	**Examination**			
	Examination Materials	1,500,000		1,500,000
Total				**18,625,000**
Year 2 Semester 2				
Income: 40 x 910,000 = 36,400,000				
1	**Allowance**			
	Teaching allowance for 270 contact hours	32,500	270	8,840,000
	Support staff allowance	5,000	300	1,500,000
2	**Examination**			
	Examination Materials	1,500,000		1,500,000
	Invigilation of six exams including refreshments	9,000	6	54,000
	Marking allowance for 240 scripts per course for 5 courses	1,500	240	360,000
	External Examiners	1,000,000		1,000,000
3	**Machine Equipment Maintenance**			
	Servicing Computers	30,000	120	360,000
	Computers Repair and maintenance	20,000	120	240,000
	Printers and Photocopy Maintenance	50,000	4	200,000

	Antivirus Purchase	50,000	10 Pieces	500,000
4	**Teaching Materials**			
	Whiteboard Markers	10,000	10 Boxes	100,000
	Chalks	5,000	2 boxes	10,000
	Board dusters	5,000	2 Boxes	10,000
	Printing and Photocopying Papers	75,000	2 Cartons	150,000
	Writing Pens	10,000	3 Boxes	30,000
5	**Administrative Cost**			
	Repair of chairs and tables	10,000	45	450,000
	Repair of locks and window fasteners	10,000	15	150,000
	Electrical Maintenances	100,000	Units	100,000
	Generator Fuel	3,500	1000 Liters	3,500,000
	Electricity Bills	600	700 units	420,000
	Water Bills	2,500	200 units	500,000
Total				**19,974,000**
Year 3 Semester 1				
Income: 40 x 910,000 = 36,400,000				
1	**Allowance**			
	Teaching allowance for 275 contact hours	32,000	275	8,800,000
	Support staff allowance	5,000	300	1,500,000
2	**Examination**			
	Examination Materials	1,500,000		1,500,000
	Invigilation of six exams including refreshments	9,000	6	54,000
	Marking allowance for 240 scripts per course for 5 courses	1,500	240	360,000
	External Examiners	1,000,000		1,000,000
3	**Machine Equipment Maintenance**			
	Servicing Computers	30,000	120	360,000
	Computers Repair and maintenance	20,000	120	240,000
	Printers and Photocopy Maintenance	50,000	4	200,000
	Antivirus Purchase	50,000	10 Pieces	500,000
4	**Teaching Materials**			
	Whiteboard Markers	10,000	10 Boxes	100,000
	Chalks	5,000	2 boxes	10,000
	Board dusters	5,000	2 Boxes	10,000
	Printing and Photocopying Papers	75,000	2 Cartons	150,000
	Writing Pens	10,000	3 Boxes	30,000
5	**Administrative Cost**			
	Repair of chairs and tables	10,000	45	450,000
	Repair of locks and window fasteners	10,000	15	150,000
	Electrical Maintenances	100,000	Units	100,000
	Generator Fuel	3,500	1000 Liters	3,500,000
	Electricity Bills	600	700 units	420,000
	Water Bills	2,500	200 units	500,000

Total				**19,934,000**
Year 3 Semester 2				
Income: 40 x 910,000 = 36,400,000				
1	**Allowance**			
	Teaching allowance for 275 contact hours	32,000	275	8,800,000
	Teaching allowance for 300 contact hours	32,000	300	9,600,000
	Support staff allowance	5,000	300	1,500,000
2	**Examination**			
	Examination Materials	1,500,000		1,500,000
	Invigilation of six exams including refreshments	9,000	6	54,000
	Marking allowance for 240 scripts per course for 5 courses	1,500	240	360,000
	External Examiners	1,000,000		1,000,000
3	**Machine Equipment Maintenance**			
	Servicing Computers	30,000	120	360,000
	Computers Repair and maintenance	20,000	120	240,000
	Printers and Photocopy Maintenance	50,000	4	200,000
	Antivirus Purchase	50,000	10 Pieces	500,000
4	**Teaching Materials**			
	Whiteboard Markers	10,000	10 Boxes	100,000
	Chalks	5,000	2 boxes	10,000
	Board dusters	5,000	2 Boxes	10,000
	Printing and Photocopying Papers	75,000	2 Cartons	150,000
	Writing Pens	10,000	3 Boxes	30,000
5	**Administrative Cost**			
	Repair of chairs and tables	10,000	45	450,000
	Repair of locks and window fasteners	10,000	15	150,000
	Electrical Maintenances	100,000	Units	100,000
	Generator Fuel	3,500	1000 Liters	3,500,000
	Electricity Bills	600	700 units	420,000
	Water Bills	2,500	200 units	500,000
Total				**19,934,000**

Appendix B Functional Fees

Table 2 shows functional fees payable to the University at the beginning of the 1st Semester of each Academic year.

Table 2: Functional fees payable to University.

Fees	**Ugandan Students (UGX)**	**International Students (USD)**	**Period**
Registration Fee Per	30,000	45	Semester
Late Registration Fee After	20,000	30	Per Semester

Two Weeks			
Examination Fee	80,000	135	Per Semester
Re-Examination Fee per Course (Retake)	6,000	10	Per Retake
Library User Fee	20,000	30	Per Semester
Capital Development Fee	80,000	45	Once
Technology Fee	50,000	60	Per Year
Internship/Field Attachment Fee	60,000	150	Per Semester
University Identity Card	20,000	15	Per Issue
Guild Fee	30,000	45	Per Semester
UNSA Fee	2,000	5	Per Year
Sports Fee	20,000	30	Per Semester
University Rules Book	5,000	5	Once
Medical Examination Fee	20,000	30	Per Semester
Medical Fee	20,000	15	Per Semester
NCHE	20,000	30	Per Year
Internship fee (Proposed)	120,000	60	Per Year