MAKERERE UNIVERSITY

COLLEGE OF COMPUTING AND INFORMATION SCIENCES
SCHOOL OF COMPUTING AND INFORMATICS TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE
P. O. BOX 7062, KAMPALA, UGANDA

REVISED BACHELOR OF SCIENCE IN COMPUTER SCIENCE (BSC CSC)

NOVEMBER 2021

Contents

1 Background

The Bachelor of Science in Computer Science programme is hosted by the Department of Computer Science in the School of Computing and Informatics Technology, College of of Computing and Information Sciences.

1.1 College of Computing and Information Sciences

The College of Computing and Information Sciences (CoCIS) was established by the University Council in 2011 by combining the then Faculty of Computing and Informatics Technology (renamed School of Computing and Informatics Technology) together with the East African School of Library and Information Science.

1.2 School of Computing and Informatics Technology

The School of Computing and Informatics Technology (SCIT) in the College of Computing and Information Sciences, Makerere University is the main centre for ICT training, research and consultancy in Uganda. SCIT grew out of the Faculty of Computing and Informatics Technology following the adoption of the collegiate system by Makerere University in 2010. The then Faculty of Computing and Informatics Technology had been earlier established by the University Council at its 100th meeting held on 15th December 2004 by upgrading the Institute of Computer Science into a Faculty with four Academic Departments, namely: Computer Science, Networks, Information Technology and Information Systems. The Institute of Computer Science that was established by the University Council in 1985 grew out of the University Computer Centre.

1.2.1 Vision

To be a leader in Computing, ICT training, and research services.

1.2.2 Mission Statement

To provide first class teaching, research, computing and ICT services responsive to national and international needs.

1.2.3 Value Statement

To be an innovative and industry-oriented School, pursuing excellence in teaching, learning, cutting edge value-added research and consultancy, community outreach, as well as providing a vibrant student life.

1.3 Department of Computer Science

The Department of Computer Science is recognized as of one of the top Computer Science departments in Africa. The Department's staff and students are actively involved in creating innovative solutions to improve people's lives. Staff and students of the Department have carried research in areas such as artificial intelligence, machine learning, computer vision and natural language processing for example automated diagnosis of both crop and human diseases, auction design for mobile commodity markets, analysis of traffic patterns in African cities, and the use of telecoms and remote sensing data for anticipating the spread of infectious diseases.

The Department of Computer Science was the pioneer department of the then Faculty of Computing and Information Technology that later evolved into the School of Computing and Informatics Technology. The Department of Computer Science is currently responsible for running programmes in the area of Computer Science. The Department currently runs the following programmes approved by University Council:

1. PhD in Computer Science
2. MSc. in Computer Science
3. Postgraduate Diploma in Computer Science
4. BSc. in Computer Science

In addition, members of the department participate actively in and support research through two research groups:

- Artificial Intelligence & Data Science Lab
- Software Systems Centre

2 Programme Description

The Bachelor of Science in Computer Science is the oldest undergraduate degree Programme in Makerere University School of Computing and Informatics Technology. It was launched at the inception of Institute of Computer Science in 2001. It was later revised in 2009 and 2013. The Bachelor of Science in Computer Science is a three year full-time programme open to all qualifying students. The programme is based on an effective curriculum, achieved through considering a number of factors that cover the opportunities, and gaps that need to be addressed by the BSC CSC programme. The Bachelor of Science Computer Science programme at Makerere University has been designed to produce graduates who are capable of creating and inventing new technological solutions that solve the most pressing problems particularly in the developing world context and contributing to their transitioning into middle-income and developed countries.

2.1 Justification/Rattionale for Reviewing the Programme

Computers are part and partial of just about every aspect of modern life and Computer Science is a core discipline that underpins the technologies of the 21st century. With the technology shifts and an economy that is based on different skills, we realized a need for revising the curriculum for our program so as to mitigate some of the obvious catastrophes and also provide the best for our graduates. We understand that for us to achieve an effective curriculum revision, it requires a thorough understanding of the processes and principles of the changing paradigms affecting curriculum development. In lieu of this, we considered various aspects in this process including:

- National Council for Higher Education (NCHE) requirement. It is a requirement by the National Council for Higher Education (NCHE) for a periodic review of curricula based on program cycles: considering that the BSc Computer Science programme was last reviewed in 2013. It has run for a cycle of three (3) years, and is due for review.

- Global Computing trends: We analysed the global computing trends and new important fields of Computer Science such as Cyber Security; Artificial Intelligence and Data Science; Machine Learning; Internet of Things; Embedded Systems; Mobile and Cloud Computing; Data Analytics; Software Engineering; new programming paradigms; tools and techniques; and others.

- Market analysis: We undertook a preliminary market analysis to evaluate the attractiveness and the dynamics of the Computer Science Market within East Africa. Special attention was paid on the kind of jobs/roles being advertised within this market segment and the different skills set that are being sought. We compared this with the Internship feedback obtained from our industrial partners where our students get posted. With this information, we were able to identify the opportunities, strengths, weaknesses and threats of our program.

- Industrial and Alumni feedback: We sought feedback from our industrial partners to generate input into this process. In particular, key stakeholders and experts from regulatory authorities, industry and academia were invited to evaluate our current program. The feedback generated in this final phase of this process was very important in reconsidering certain content of the courses we have been offering.

- Student sample: A sizable sample of our students was also sampled. In here, we were testing their understanding of what we are offering against what they expected. These respondents pointed us to various areas which they considered a duplication of effort or redundancy in the program. Their feedback was thus important in helping us to eliminate duplicates and also for strengthening certain areas of our program.

- External examiners' and Quality assurance reports: These reports were also a great input into this review process. Although some of the comments given were majorly on assuring quality of students' learning experiences and also the fair assessment, other comments given to improve the program were very significant in highlighting areas which were found lacking in our program.

2.2 Programme Objectives

The Bachelor of Science in Computer Science programme aims to educate and develop students in strong theoretical and practical foundations of Computer Science to enable them excel in the workplace and to be lifelong learners. It is designed to equip students with capabilities for creating and inventing new technological solutions that solve the most pressing problems particularly in the developing world context. The programme is designed to address the diverse needs of multiple stakeholders by focusing on Computer Science skills required to succeed in the 21st century. Specifically, the objectives of the programme are to:

- Develop professionals with theoretical and practical skills in Computer Science;
- Strengthen institutional capacity and building in Computer Science in tertiary institutions, the private and public sector
- Build capacity with a practical orientation needed to link up the Computer Science sector with Government and Industry under the broader perspective of Information and Communication Technology (ICT)

2.3 Program Learning Outcomes

Graduates of the BSc. Computer Science will be trained with the following learning outcomes.

At the end of this program, graduates of the BSc. Computer Science should:

- Be able to have theoretical and practical skills in Computer Science;
- Have the capacity as ICT professionals in Computer Science to meet needs in tertiary institutions, the private and public sector.
- Have the capacity with a practical orientation needed to link up the Computer Science sector with Government and Industry under the broader perspective of Information and Communication Technology (ICT) and

2.4 Value Proposition

The Makerere University Computer Science programme is distinct from other computing & IT degrees elsewhere in that it does not train students to just do maintenance, management and administration of IT systems, rather the Bachelor of Computer Science programme is designed to equip students with capabilities for creating new knowledge and innovations that advance the current technologies.

2.5 Target Group

The programme targets two categories of people. These are A level leavers and diploma holders in relevant disciplines.

2.6 Nature of the Programme

The programme has both government and privately sponsored students.

2.7 Duration

The duration of this program is three (3) academic years consisting of six semesters and two recess terms in the first and second year. Each semester lasts seventeen (17) weeks. Each recess term lasts ten (10) weeks.

2.8 Tuition Fees

Tuition fees for privately sponsored students shall be 3,024,000 Uganda Shillings per year for Ugandans and East Africans and 4,536,000 Uganda Shillings per year for non-Ugandans. Recess Term fees for Year 1 Recess shall be 200,000 Uganda Shillings.

2.9 Who is a Computer Science Graduate?

Our interdisciplinary approach to study sets us apart from traditional, highly structured computer science departments. As such, our students gain a strong foundation in computer science while preparing themselves to use their degree to solve real-world challenges. The objective of the computer science program is to provide computer science graduates with a balanced breadth and depth of knowledge in computer science that allows them the choice between continuing their education in graduate school and beginning their professional career, and to excel in either environment. Therefore the computer science graduate should be able to achieve the following outcomes:

- ability to apply knowledge of mathematics and science to carry out analysis of computer science problems and design appropriate solutions

- ability to use techniques, skills, and modern software development tools necessary for computing practice
- ability to identify, formulate, and solve computer science problems
- ability to design a computing system to meet desired needs
- ability to apply problem-solving strategies to new, unknown, or open-ended situations in computer science
- knowledge and understanding of the impact of the many sub-disciplines of computer science
- ability to function on teams
- ability to use written and oral communication skills effectively
- understanding of professional and ethical responsibility
- recognition of the need for and ability to engage in lifelong learning
- ability to acquire and use the ever-changing technical knowledge required of computing professionals

2.10 Career Prospects

The Computer Science programme prepares graduates for many successful careers in different environments including technology industry, government, academia, private and business organizations. The list below provides some of the possible career options for a graduate of the BSc. in Computer Science:

- Create IT/Software Startups
- Computer/Cyber Security
- Software Engineers/Computer Programmers
- Data Scientist
- Data Engineer
- ICT Project Consultants

- Systems Security Analyst
- Systems Analyst, and Business Intelligence Analyst
- Database, Systems and Network Administrators
- Artificial Intelligence Software Engineer
- Embedded Systems Engineer

3 The Programme and Regulations

3.1 Admission Requirements

To be admitted to the B.Sc (Computer Science) program, a candidate must satisfy the general admission requirements for Makerere University. The following are the available entry schemes for admission:

3.1.1 Direct Entry

Candidates seeking admission through this avenue must have obtained:-

- At least a principle pass in Mathematics in the Uganda Advanced Certificate of Education (UACE) or its equivalent
- At least two principle passes at the same sitting in UACE in any of the following subjects: - Economics, Entrepreneurship, Geography, Physics, Technical Drawing, Chemistry and Biology.
- A minimum weighted point set by the Makerere University Admissions Board.

For purposes of computing weighted points, the A' level subjects shall be grouped and weighted as per the University weighting system as illustrated in the table below.

Group	**Weight**	**Subjects**
Essential	3	Mathematics and any other best done subject among: Physics, Chemistry, Biology, Economics, Technical Drawing
Relevant	2	Any other best done subject of all A' level subjects.
Desirable	1	General Paper, Subsidiary Mathematics, Subsidiary ICT

3.1.2 Diploma Entry

For a candidate to be admitted via the diploma scheme, he/she must:

1. Have at least 5 passes got at the same sitting of Uganda Certificate of Education or its equivalent, with credits in English and Mathematics
2. Have at least 1 principal pass and 2 subsidiary passes from the same sitting of the Uganda Advanced Certificate of Education (UACE) or its equivalent
3. Have a Honours Diploma in Computer Science, Information Technology, Computer Engineering, Mathematics, Electrical Engineering or Software Engineering from an Institution recognized by the National Council for Higher Education in Uganda. Students admitted via the diploma scheme shall start in first year.

3.1.3 Degree Holders

For a candidate to be admitted via the degree holder scheme, s/he must; Satisfy the general minimum entry requirements of Makerere University. In addition candidates seeking admission through this avenue must have obtained:

1. At least second class degree in line of Mathematics, Statistics, Engineering, Computing, Physics, Chemistry, Education, Economics, Geography
2. The degree must be attained from an Institution recognized by the National Council for Higher Education in Uganda.

3.2 Progression

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

3.2.1 Normal Progress

This occurs when a student passes each course taken with a minimum Grade point of 2.0.

3.2.2 Probationary Progress

This is a warning stage and occurs if either the cumulative grade point average (CGPA) is less than 2.0 and / or the student has failed a core course. Probation is waived when these conditions cease to hold.

3.2.3 Discontinuation

When a student accumulates three consecutive probations based on the CGPA or the same core course(s), he / she shall be discontinued.

3.2.4 Retaking a Course

A student may re-take any course when it is offered again in order to pass if the student had failed the course. A student may take a substitute elective, where the student does not wish to re-take a failed elective.

3.3 Weighting System

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

3.4 Semester Load and Minimum Graduation Load

The normal semester load is between 18 and 20 credit units. The minimum graduation load is 124 credit units.

3.5 Course Assessments

Each course will be assessed on the basis of 100 marks with proportions as follows:- Coursework 40% and Examination 60%. A minimum of two course assignmentstests shall be required per course. In some courses examinations may be practical based (e.g., a practical project). Course work may consist of individual tests, field study, practical projects, group assignment and presentations in each semester.

3.6 Grading of Courses

Each course will be graded out of a maximum of 100 marks and assigned an appropriate letter grade and a grade point. Courses given for each Program shall be appropriately weighted and scores in students' Grade Point Average (GPA) shall be a result of assigned grades as set out in the following table using a five point rating system in averaging the final grade of a graduate.

Marks	**Letter Grade**	**Grade Point (GP)**
90 - 100	A+	5.0
80 - 89	A	5.0
75 - 79	B+	4.5
70 - 74	B	4.0
65 - 69	C+	3.5
60 - 64	C	3
55 - 59	D+	2.5
50 - 54	D	2.0
45 - 49	E	1.5
40 - 44	E-	1.0
Below 40	F	0.0

The following additional letters will be used, where appropriate: W Withdrawal from Course;

I Incomplete;
AU Audited Course Only;
P Pass;
F Failure.

3.7 Minimum Pass Mark

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

3.8 Calculation of Cumulative Grade Point Average (CGPA)

Calculation of Cumulative Grade Point Average (CGPA)
The CGPA shall be calculated as follows:-

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

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

3.9 Classification of Award for Degrees

The classification of awards of degrees shall be as set out in the following table and inline with the University regulations.

Class of Degree	**CGPA Range**
First Class	4.40–5.00
Second class (Upper Division)	3.60–4.39
Second class (Lower Division)	2.80–3.59
Pass	2.00–2.79

4 The Revised Curriculum

4.1 Main Changes in the Revised Curriculum

The major changes in the revised curriculum include:

(a) Twelve (12) new courses were developed to cater for the new technology trends and/or improve the breadth and depth of the Computer Science graduates.

(c) Structured Programming and Object-oriented have been merged into one course to provide a comprehensive introduction to programming using a modern programming language that combines the two paradigms.

(d) Basic ICT introduction course Computer Literacy has been made an audited course. By the Admission requirements, all A'level students who attempt Mathematics take ICT studies. Thus all the students admitted have basic computer literacy. Moreover all Computer Science courses involve usage of Computers where students can improve their basic ICT skills.

(e) New courses such as robotics, machine learning, embedded systems, cloud computing, Digital innovation and computational thinking, probability and statistics, have been introduced in the revised curriculum to cater for new and emerging knowledge areas in Computer Science.

(f) Mathematical courses have been synthesised and merged into a few to minimise the repetition of A'level mathematics and focus on the providing the right dose of mathematics that is necessary for Computer Scientists.

(g) Course content has been revised into modular units that can be assessed independently. Further more, the assessment methods have been diversified to include regular quizzes, practical mini projects, and project-based assessments.

(h) The model of delivery has been expanded to include use of blending learning approaches combining class and online lectures. All courses are expected to be developed as interactive online courses on the Makerere University e-Learning Platform to facilitate student-centered learning.

4.2 The Curriculum

The curriculum is broken down into a set of core courses that each student is required to undertake in a semester and a set of electives that students are required to select from to obtain the minimum semester load. The details of the course structure are shown below

Key

1. **Modified**: This remark means that the course unit has been revised by content.
2. **New**: Means the course unit has been freshly introduced in the department
3. **Old**: Means no change on the course units name, code and content.
4. **LH**: Means Lecture Hour.
5. **PH**: Means Practical Hour.
6. **CH**: Means Contact Hour.
7. **CU**: Means Credit Unit.

4.2.1 Year I, Semester I

Code	**Course Name**	**LH**	**PH**	**CH**	**CU**	Remark
CSK 1101	Communication Skills	45	30	60	4	Old
CSC 1102	Structured & Object-Oriented Programming	30	60	60	4	New
CSC 1104	Computer Architecture & Organisation	30	60	60	4	Modified
CSC 1105	Mathematics for Computer Science	30	60	60	4	New
CSC 1109	Digital Innovation & Computational Thinking	30	30	45	3	New
Audited Course						
CSC 1100	Computer Literacy	30	60	60	4	Old
TOTAL					**19**	

4.2.2 Year I, Semester II

Code	**Course Name**	**LH**	**PH**	**CH**	**CU**	Remark
CSC 1200	Operating Systems	30	60	60	4	Modified
CSC 1201	Probability & Statistics	30	60	60	4	New
CSC 1202	Software Development Project	15	90	60	4	Modified
IST 1204	Systems Analysis and Design	30	60	60	4	Old
CSC 1204	Data Structures and Algorithms	30	60	60	4	Old
TOTAL					**20**	

4.2.3 Year I, Recess Term

Code	**Course Name**	**LH**	**PH**	**CH**	**CU**	Remark
CSC 1303	Cisco Certified Network Associate (CCNA) **Audited**	45	60	75	5	Old
CSC 1304	Practical Skills Development	15	90	75	5	Modified
TOTAL					**5**	

4.2.4 Year II, Semester I

Code	**Course Name**	**LH**	**PH**	**CH**	**CU**	Remark
CSC 2105	Discrete Mathematics	30	30	45	3	Modified
BSE 2106	Computer Networks	30	60	60	4	Old
CSC 2107	Database Management Systems	30	60	60	4	New
CSC 2114	Artificial Intelligence	30	60	60	4	Modified
CSC 2118	Embedded and Real-time Systems	30	60	60	4	New
TOTAL					**19**	

4.2.5 Year II, Semester II

Code	**Course Name**	**LH**	**PH**	**CH**	**CU**	Remark
IST 2203	Research Methodology	30	60	60	4	Old
CSC 2206	Introduction to Machine Learning	30	60	60	4	New
CSC 2209	Cloud Computing	45	30	60	4	New
CSC 2210	Automata, Complexity and Compatibility	30	30	45	3	Modified
	Electives (select 1)					
CSC 2207	Robotics	30	60	60	4	New
CSC 2208	Software Quality and Verification	30	60	60	4	New
CSC 2218	Software Construction	30	60	60	4	New
TOTAL					**19**	

4.2.6 Year II, Recess Term

Code	**Course Name**	**LH**	**PH**	**CH**	**CU**	Remark
CSC 2303	Field Attachment	-	300	75	5	Old
TOTAL					**5**	

4.2.7 Year III, Semester I

Code	**Course Name**	**LH**	**PH**	**CH**	**CU**	Remark
BAM 2102	Entrepreneurship Principles	30	30	45	3	Modified
CSC 3115	Advanced Programming	30	60	60	4	Old
CSC 3118	Computer Science Project I	-	150	75	5	Modified
CSC 3119	User Interface Design	45	30	60	4	Modified
	Electives (select 1)					
IST 3110	Business Process Management	30	60	60	4	Old
CSC 3114	Cryptology and Coding Theory	45	-	45	3	Modified
CSC 3117	Operations Research	30	30	45	3	New
CSC 3121	Computer Graphics	30	30	45	3	Modified
IST 3208	Modelling and Simulation	30	30	45	3	Old
TOTAL					**19**	

4.2.8 Year III, Semester II

Code	**Course Name**	**LH**	**PH**	**CH**	**CU**	Remark
CSC 3205	Compiler Design	30	30	45	3	Old
CSC 3207	Computer Security	30	30	45	3	Modified
CSCS 3211	Computer Science Project II	-	150	75	5	Old
CSC 3217	Emerging Trends in Computer Science	30	30	45	3	Old
	Electives (Select 1)					
BSE 2206	Data Communications	30	60	60	4	Old
BSE 3202	Distributed Systems Development	30	60	60	4	Modified
BIS 3205	Data Warehousing and Business Intelligence	30	60	60	4	Old
TOTAL					**18**	

5 Detailed Curriculum

5.1 Year I Semester I

5.1.1 Communication Skills

Course Name:	Communication Skills
Course Code:	CSK 1101
Course Credit:	4
Contact Hours:	60
Year of Study:	1
Semester:	1

(a) Course Description
This course provides students with skills of effective communication. Emphasis is provided on communications in todays business environment that has increasingly been shaped by Information technology. Students will be taught how to effectively communicate technical information to lay audiences using oral, written and non-verbal communication.

(b) Course Objectives
The objectives of the course are:

- Ensure that students develop and apply effective communication and writing skills in management of information systems.
- To familiarize students with the concept and Effective communication, reports and business documents required effectively in the management of information systems.
- To teach students the principles of designing formal and informal reports to meet the needs of a variety of information system managers.
- To enable students understand the importance of effective organization and personal communication.
- To enable students prepare some business documents in the organization / institutions.
- To provide tools and strategies with which students can systematically improve their communication skills.

(c) Learning Outcomes
By the end of the course unit, the student should be able to:

- Identify, explain, and demonstrate different types of communication techniques.
- Organise and present ideas individually and as part of a team.
- Demonstrate interpersonal skills including business etiquette, active listening, team participation, and leadership skills.
- Identify ways of communicating effectively through an appropriate balance between interpersonal and technical capabilities.
- Produce a text-based report or proposal, using Microsoft Word.
- Apply a universal process that enables you to solve communication problems now and throughout your entire carrier

(e) Detailed Course Content

- **Module 1 Writing Skills (20 Hours):** Thinking critically/ selectively before the writing process; selecting the relevant details; organizing the relevant details logically; Writing the reports essays, letters and taking notes in appropriate register; Avoiding ambiguities, fallacies, irrationalities; Providing supportive evidence; Editing documents, proof reading; Writing and expanding information; Quoting and citing references; Writing a curriculum vitae.
- Module 2: **Reading Skills (20 Hours):** The use of skimming; scanning inference and prediction in reading; Intensive and critical reading; Acquisition of specific reading skills; Interpretation of non linear texts; Locating information and comprehension.
- **Module 3: Speaking and Listening Skills to Enhance Effective Public Relations (20 Hours)**: The art of persuasion in effective speaking; Conducting interviews; Conducting meetings; Participating in group discussions and tutorials; Non verbal communication clues; Presentation seminars, seeking clarification etc.; Expression of politeness; Public speaking; Proper listening skills.

(f) Study Materials
Text books, conference and journal publications, and online resources.

(d) Mode of Delivery
This course will involve online or face-to-face class lectures blended with practical sessions. Students will be introduced real life examples and illustrations

to enable them apply the acquired knowledge with the business context. Students will be required to analyse business domain problem and come up with communications to address the needs as part of their coursework assessment

(g) Mode of Assessment
Course work assessment will contribute (40%) and examination will contribute (60%).

(h) Reading List

[1] COURTLAND L. BOVE and JOHN V. THILL, Business Communication Today. 8th edition. *Upper Saddle River, NJ: Prentice Hall International Inc.*, (Call No. HF5718 Bov2005).

[2] Bennie, B.. 101 ways to improve your communication skills instantly 4th edition,2005

[3] Peggy, K.. The hard Truth About soft skills: Work Place Lessons Smart People wish they had learned sooner, 2008.

[4] Grant, A.E. Meadows, Focal J.H.. Communication Technology Update and Fundamentals ,Focal Press, 2008

5.1.2 Structured and Object-Oriented Programming

Course Name:	Structured and Object-Oriented Programming
Course Code:	CSC 1102
Course Credit:	4
Contact Hours:	60
Year of Study:	1
Semester:	1

(a) Course Description

The course is to create a strong foundation in the principles and practice of modern computer programming. It should give an in depth understanding of structured and Object Oriented programming. It is to cater for both Structured and Object Oriented programming (O-O) topics like control structures, objects and classes, encapsulation, inheritance, polymorphism, interfaces, and exception handling. A state of the art programming language like Python or Java is to be used. Students are to cover both theoretical principles and hands on practical skills.

(b) Course Objectives
The aims of the course are to provide the student with:

- Introductory knowledge about structured programming
- Knowledge and techniques of evaluating syntactic and semantic correctness of a computer program
- Knowledge in planning, organization, and development of complete computer applications
- Strong practical basis in object-oriented style of programming

(c) Learning Outcomes

Students who successfully complete this course of study will be able to:

- Understand the basic terminology used in computer programming
- Write, compile and debug programs in a structured and O-O manner (like Python)
- Design programs involving decision structures, loops and functions
- Explain the principles of the object oriented programming paradigm specifically including encapsulation, inheritance and polymorphism
- Design, develop, test, and debug programs using object oriented principles in conjuncture with an integrated development environment
- Describe and explain the factors that contribute to a good object oriented solution, reflecting on your own experiences and drawing upon accepted good practices

(e) Detailed Course Content

- Module 1 (16 hours): **Structured Programming**
  - Program structure
  - Branching and Iterations: Control flow statements
- Module 2 (16 hours): **Data manipulation**
  - Functions

    - Strings, pointers, and arrays
  - Module 3 (28 hours): **Object-Oriented Concepts**
    - Object-Oriented programming: Classes and objects
    - Inheritance and visibility modifiers
    - Interfaces and abstract classes
    - Exception handling

(d) Mode of Delivery

There will be by blended learning including online or face-to-face class sessions, tutorials, and/or practical sessions.

(g) Mode of Assessment

The assessment will be done by tests and take home practical assignments (40%), practical examination (30%) and written examination (30%)

(h) Reading List

1. Brian W. Kernighan, Dennis M. Ritchie: C Programming Language; Prentice Hall, 2000
2. Patrick Naughton & Herbert Schildt: The Complete Reference – Java 2, Fifth Edition.
3. Problem Solving with Algorithms and Data Structures, *Release 3.0* , by Brad Miller, David Ranum, 2013

5.1.3 Computer Architecture & Organisation

Course Name:	Computer architecture & Organisation
Course Code:	CSC 1104
Course Credit:	4
Contact Hours:	60
Year of Study:	1
Semester:	1

(a) Course Description
This course introduces the logical architecture and organization of computer systems. It highlights the lower end operations in a typical computer as well as the way computers manage their resources during operation. The course opens up a student to be an informed user of the computer rather than a passive recipient of the computer services.

(b) Course Objectives
The aims of the course are:

- To provide students with a fundamental understanding of the functional components of a computer system, and how they are organized.
- To highlight to students the way computers process and store the data.
- To highlight internal management issues in computer systems.

(c) Learning Outcomes
By the end of the course a student should be able to:

- Explain the difference between Computer Organization and Computer Architecture;
- Understand the organisation of a computer system in terms of its main components
- Discuss how information is processed in the computer from the time it is input to the time of output;
- Explain how the Micro processers operate;
- Understand the operation of electronic logic elements
- Understand different processor architectures
- Understand input/output mechanisms
- Understand the various parts of a system memory hierarchy

(e) Detailed Course Content

- **Module 1: Introduction to computers and Data Representation: (12 CH)**
  Understanding of Organisation & Architecture, Function & structure of computer Components, Technology Trends, Instruction set Architectures, Measuring CPU performance.

Integer Formats, Binary, Octal and Hexadecimal Systems, Negative integers and 2s Complement, Floating Point Formats, BCD Formats, Alphanumeric. Hardware Codes.

- **Module 2: Digital and Sequential circuits (24 CH)**
  Basic Digital Circuits: Logic gates, Karnaugh maps, Combinatorial Circuits, Binary Adders, Multiplexers and Demultiplexers, Comparators, Decoders and Encoders, Code Converters, ROMS and PLAs, Sequential Circuits, Flip Flops and Latches, R-S flip flops, J-K flip flops, T flip Flops, D flip flops, Registers, Shift Registers and Data Transmission, Sequential Network Design.
- **Module 3: Assembly Level Machine Organization (12 CH)**
  Basic Organisation of the Von Neumann machine, Control Unit, Instruction fetch, decode & execution, Instruction sets and types (Data manipulation, control, I/o), Assembly/machine language programming, instruction formats,Register Organisation and Addressing modes, sub-routine call and return mechanisms, I/O & interrupts.
- **Module 4: Micro Computer Architecture:(12 CH)**
  Architecture of the CPU, Memory(cache,internal and external) and memory management, I/O Devices and Interfaces, System Bus, Examples of CPU Structures, Machine Language Instructions, Instruction Formats and Addressing Modes.
  The Processing Elements: Macroinstruction execution, Internal Bus Transfers, Detailed Internal Architecture, Microcontrol: Hardwired Control, Microprogrammed Control, Reduced Instruction Set Computers.

(d) Mode of Delivery
Teaching will be in terms of blended learning including lectures as well as tutorials.

(g) Mode of Assessment
Assessment will be in form of tests and assignments (40%) and final examination (60%)

(h) Reading List

[1] William Stallings, Pearson Education. Computer Organization and Architecture (Designing for Performance), 09th edition, (2012).

[2] Glenn A. Gibson. Computer Systems; Concepts and Design:

5.1.4 Mathematics for Computer Science

Course Name:	Mathematics for Computer Science
Course Code:	CSC 1105
Course Credit:	4
Contact Hours:	60
Year of Study:	1
Semester:	1

(a) Course Description

The course gives a foundation and deep understanding of mathematical skills that are necessary in studying advanced aspects of computer science. The course extends the mathematics that was taught in high school but this time focusing on areas that will help the candidates conceptualise computer science knowledge like in image processing, machine learning, graphics, interface design , robotics and security.

(b) Course Objectives

The objectives of the course are to equip students with skills on

1. Finding limits for functions
2. Establishing continuity of functions
3. Evaluating finite and infinite sequences and progressions
4. Performing operations of Matrix algebra
5. Analysing and interpreting vector spaces
6. Performing plane arithmetic

(c) Learning Outcomes

By end of the course, candidates should be able to

1. Compute limits of polynomial, logarithmic, exponential, trigonometrical and mixed functions
2. Establish continuity and diferentiability of functions
3. Calculate inverse, determinant and transpose of matrices
4. Perform 2D and 3D transformations using matrices
5. Perform Vector and vector space algebra
6. Generate mathematical characteristics in regard to vectors and planes

(e) Detailed Course Content

- **Module 1: Limits and Continuity (10 CH)**
  This module addresses calculation of limits of simple functions as well as fairly complex functions (like composite polynomials and exponential functions). It involves the ϵ, σ, calculation of limits of functions using limit laws (eg Squeeze theorem, L'opitals rule, etc). It also involves establishment of limits to infinity and horizontal asymptotes. It also covers differentiability of functions using continuity tests as well as generating derivatives of functions, applying chain rules and application of derivatives into real life applications like rates of change.
- **Module 2: Progressions and Sequences (10CH)**
  This module covers computation of standard progressions (AP and GP), special progressions and sequences as well as computing their limits and Mathematical upper / lower bounds (where they exist). Techniques like breaking sequences into separate progressions will be emphasized.
- **Module 3: Matrix Algebra (10 CH)**
  This module will address the fundamental skills in matrix algebra (matrix algebra rules, transposes, inverses, determinants, egine vectors and kernels) as well as how matrices are used in mathematical operations like transformations in vector spaces and solving of equations.
- **Module 4: Vectors and Vector Spaces (15 CH)**
  This module will cover computations in vectors and vector spaces, vector operations, linear combination of vectors, spanning sets , linear dependence and Independence and orthogonality.
- **Module: 5 Vectors and Planes (15 CH)**
  This module will cater for the analytical interpretation of vectors and planes. This will include vector equations of lines in n- space, equations of planes, as well as characteristics between planes and vectors like angle between them, distance between them, etc

(f) Study Materials
Study materials will include textbooks, prepared tutorials, group discussions and online webinars

(d) Mode of Delivery
The study mode will be a blend of learner centered delivery via online or face to face classes. The teacher will introduce the topic and guide students on where

further reading can be done. Student centered activities like discussion forum, will be employed to supplement teachers work.

(g) Mode of Assessment

1. Each module will be assessed by at least one assignment and one test. The composite scores shall constitute progressive assessment (40%)
2. There shall be a final exam at the end of the semester that shall constitute 60% of the final score

(h) Reading List

[1] Robert A. Beezer (2015) A First Course in Linear Algebra http://linear.ups.edu/html/fcla.html

[2] Crowell and Slesnick(2008) Calculus with Analytic Geometry

5.1.5 Digital Innovation & Computational Thinking

Course Name:	Digital Innovation & Computational Thinking
Course Code:	CSC 1109
Course Credit:	3
Contact Hours:	45
Year of Study:	1
Semester:	1

(a) Course Description
Computer Science is central to digital innovations that solve some of the world's most challenging problems. This course aims to introduce students to the world of computational thinking and its application to problem solving, process of digital solution development, innovation ecosystem and entrepreneurship.

(b) Course Objectives
The objectives of this course are to:

1. equip students with core computational and problem solving skills in computer science including including logical, algorithmic, and abstraction.
2. develop knowledge and understanding of digital innovation and entrepreneurship processes and systems

3. development of business models for digital innovations
4. skills to present complex computational innovations in an easy to understand format

(c) Learning Outcomes
By the end of this course students should:

1. demonstrate capability of computational and analytical thinking and problem solving as applied to real world problems
2. have knowledge and understanding of digital innovation and entrepreneurship processes and systems
3. be able to use problem-solving strategies to develop computer science solutions to problems.
4. be able to develop business models for digital innovations and have acquired skills to present complex computational innovations in an easy to understand format.

(e) Detailed Course Content

- **Module 1: Computational thinking (06 Hours)**. Introduction to what is computational thinking, logical and algorithmic thinking. Deductive and inductive reasoning, Boolean logic and its importance to computation thinking.
- **Module 2: Problem solving and decomposition (15 Hours)**: Problem definition, decomposition, abstraction, patterns and generalisation. Abstraction and modelling. Anticipating and dealing with errors. Evaluation of a solution for correctness, efficiency elegance and usability.
- **Module 3: Application of computational thinking to software development (09 Hours)** Computational thinking in software development using a programming language like Python. Creating, finding and using abstractions and patterns in programming, testing and evaluating programs, forming generalisations and abstractions,
- **Module 4: Digital Innovation thinking (15 Hours)** , principles, processes and sources of innovative opportunities. Entrepreneurial thinking, processes and strategies. Innovation management, communication and pitching.

(f) Study Materials
The study materials will include a wide range of digital resources including textbooks, technology reports and articles, videos, and descriptions of real world case studies.

(d) Mode of Delivery
The teaching pattern is by blended learning including lectures, tutorials, practical lab work, self-study, case studies, use of e-learning management system, group discussion and class presentations

(g) Mode of Assessment
Course work (40%)and exam (60%) will include review of real-world challenges and translating them into solutions and business models, review of technology articles and pieces, quizzes, mini projects, and final project.

(h) Reading List

[1] Karl Beecher Computational Thinking: A beginner's guide to problem-solving and programming, BCS Learning and Development Ltd, 2017.

[2] David Riley and Kenny A. Hunt Computational Thinking for the Modern Problem Solver, Chapman and Hall/CRC, 2014.

[3] Paolo Ferragina and Fabrizio Luccio Computational Thinking: First Algortihms, Then Code, Springer, 2018.

[4] Guttag, John. Introduction to Computation and Programming Using Python: With Application to Understanding Data Second Edition. MIT Press, 2016.

[5] Bessant John, and Tidd Joseph. Innovation and entrepreneurship. Wiley & Sons, 2015.

[6] Osterwalder A., andPigneur Y.. Business model generation: A handbook for visionaries, game changers, and challengers. Hoboken, NJ: Wiley. ISBN-13: 978-0-470-87641-1, 2010.

[7] Wise, S. E. Hot or not: How to know if your business idea will fly or fail?. Toronto, ON: Ryerson Entrepreneur Institute. ISBN-13: 978-1-46802-449-4, 2011.

5.1.6 Computer Literacy

Course Name:	Computer Literacy
Course Code:	CSC 1100
Course Credit:	4
Contact Hours:	60
Year of Study:	1
Semester:	1

(a) Course Description

In this course, students are to learn about the basic organization, concepts and terminologies in a computerized environment. They are also to get an in depth understanding of common computer applications. The use of related applications in different operating systems will be explored.

(b) Course Objectives

The aims of the course unit are to:

- Equip students with basic knowledge about computer organization;
- Equip students with skills of using common office applications;
- Expose students to different operating systems;
- Equip students with skills of how to use the Internet;
- Equip students with knowledge about common text editors in different operating systems.

(c) Learning Outcomes

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

- Describe the different parts of a computer;
- Describe the historical evolution of computers;
- Competently use the common office applications in at least two operating systems and;
- Competently know how to use the Internet.
- Competently use common text editors in at least two operating systems.

(e) Detailed Course Content

- General computer organization (12 Hours)
- Historical perspectives of computing (8 Hours)
- Common Microsoft office packages (12 Hours)
- Office packages in other operating systems (8 Hours)
- Text editors (4 Hours)
- Common Command Line commands (Linux, Dos) (8 Hours)
- Using the web (4 Hours)
- Understanding the Internet (4 Hours)

(d) Mode of Delivery
Blended learning including:

- Lectures
- Practical sessions in the computer laboratories.
- Online learning management systems
- Class discussions and presentations
- Problem-based/case studies
- Project-based assignments that can be done in groups

(g) Mode of Assessment
The assessment will be in form of:

- Tests 20%
- Individual assignment 20%
- Final written exam 60%

(h) Reading List

[1] Preston, J. Preston, S. & Ferrett, R. Computer Literacy: Computing Fundamentals, 2nd Edition. Prentice Hall Press., 2013.

[2] Williams, B.K. & Sawyer, S.C., Using Information Technology; A practical introduction to computers & Communications. Richard D Irwin, 2007

[3] Snyder, L. 2018. Fluency with information technology: Skills, Concepts, and Capabilities, 4th Edition. NY, NY : Pearson, 2018

5.2 Year I Semester II

5.2.1 Data Structures and Algorithms

Course Name:	Data Structures and Algorithms
Course Code:	CSC 1204
Course Credit:	4
Contact Hours:	60
Year of Study:	1
Semester:	2

(a) Course Description
The course gives students a firm foundation of data structures and algorithms. The course trains students on systematic development and analysis of algorithms. The importance of algorithm complexity on computer performance is emphasized. Typical computational problems and their solutions/analysis are to be covered.

(b) Course Objectives
The objectives of this course are to educate the students in:

- How to use various data structures while developing in a particular programming language as well as how to implement some of the most common algorithms used with such data structures.
- Define and explain advanced data types such as stacks, queues, lists, trees and graphs; write programs using them.
- Define, discuss and explain the main algorithms and techniques (such as sorting, searching, hashing, traversal and recursion) and write programs using these algorithms.

(c) Learning Outcomes
By the end of the course, the student should be able to

- Appreciate the role of data structures and algorithms in computer programs.
- Describe a step by step analysis and design of computer algorithms.
- Analyze generic algorithmic problems and apply them to computational scenarios.

(e) Detailed Course Content

- **Module 1: Elementary data structures (6 CH)** the need for data structures; the role of algorithms in computing, linear and non-linear structures; Abstract Data Types.
- **Module 2: Recursive data structures (14 CH)** recursion; lists, stacks, queues, dequeues, linked lists and their implementation; Trees: binary trees, balanced trees.
- **Module 3: Sorting Algorithms (12 CH)** Simple: bubble sort, insertion sort, selection sort; Fast: Mergesort, Heap sort, Quick sort
- **Module 4: Storing and Searching (8 CH)** hash tables, binary search trees, AVL trees.
- **Module 5: Graph traversals (8 CH)** Breadth-First Search and Depth-First Search; shortest path, spanning trees
- **Module 6: Algorithm analysis (8 CH)** Asymptotic analysis; Big-O, Omega and Theta notations, orders of growth, worst case, average case, best case and amortized analysis
- **Module 7: Algorithm design (4 CH)** Dynamic algorithms, Divide-and-conquer algorithms, Greedy algorithms.

(d) Mode of Delivery
The teaching pattern is by blended learning including lectures and tutorials, practical lab work, group discussions and class presentations.

(g) Mode of Assessment
The assessment will be done by tests and assignments (40%) and final examination (60%)

(h) Reading List

[1] Mark Allen Weiss Data Structures and Algorithm Analysis in C++, 4th Edition, 2014

[2] Michael T. Goodrich and Roberto Tamassia and Michael H. Goldwasser. Data Structures and Algorithms in Python, 2013.

[3] Alfred V. Aho and John E. Hopcroft and Jeffrey D. Ullman. Data Structures and Algorithms. Addison-Wesley, 1983.

[4] Thomas H. Cormen and Charles E. Leiserson and Ronald L. Rivest and Clifford Stein. Introduction to Algorithms, Second Edition, 2001.

5.2.2 Probability & Statistics

Course Name:	Probability & Statistics
Course Code:	CSC 1201
Course Credit:	4
Contact Hours:	60
Year of Study:	1
Semester:	2

(a) Course Description

This course strengthens the student's command of probability and statistics. It is meant to lay ground for the application of probability and statistics concepts in the general field of Computer Science. The course prepares candidates to understand the concept and rules of probability as well as get an in depth understanding and manipulation of statistical distributions (Discrete and Continuous). The course also prepares students to use and apply descriptive statistics and statistical inference to answer questions about real-world data. Students will Students will learn how to use statistical computing packages available R and Python such as Numpy, which are popular in Data Science.

(b) Course Objectives

The objectives of the course are to equip the students with skills to

1. Generate probabilities of events or sequences of real life events under given conditions
2. Solve probabilistic and statistical problems related to probability distributions
3. Generate representations of distributions real life scenarios and compute related probabilities
4. Solve compounded distributions' problems
5. Analyse data using descriptive statistics
6. Apply statistical Inference and regression methods to real world examples.

(c) Learning Outcomes

By the end of the course, the candidate shall be able to

1. Solve probability problems using common approaches and tools (probability trees, contingency table, bayes theorem, etc)

2. Formulate and solve mean, variance and instance probability problems for discrete and continuous random variables
3. State distributions and solve distribution based probabilities for common special discrete and continuous random variables
4. Solve problems related to combined distributions including conditional distributions
5. Gained knowledge of descriptive statistics and statistical Inference, and regression methods for data analysis and use of statistical computing packages and libraries.

(e) Detailed Course Content

- **Module 1:** Introduction to Probabilities and Random Variables (09 CH) Probability of an even, probability experimentation, generation of probabilities of multiple or repeated events using probability trees, generation of probabilities of combined events, conditional probabilities and Bayes Theorem and Independent events.
- **Module 2:** Discrete Random Variables and Distributions (06 CH) General Discrete Random variables, characteristics, Expectance and Variance of discrete random variables, special discrete random variables: Bernoulli Distribution, Binomial Distribution, Geometric Distribution, Negative Binomial Distribution, Hypergeometric Distribution, Poisson Distribution.
- **Module 3:** Continuous Random Variables and Distributions (06 CH) General Continuous Random Variables, expectance and variance of continuous random variables, special continuous random variables: Uniform Distribution, Gamma Distribution, Beta Distribution, Normal Distribution, Lognormal Distribution, Inverse Gaussian Distribution, Logistic Distribution.
- **Module 4:** Probability Moments and Chebychevs inequality (06) Moments of Random Variables, Expected Value of Random Variables using moments, Variance of Random Variables using moments, Moments for validation of random variables, Chebychev Inequality, Moment Generating Functions
- **Module 5:** Bivariate and Conditional Random variables (09 CH) Bivariate Discrete Random Variables, Bivariate Continuous Random Variables, Conditional Distributions, Independence of Random Variables (Covariance)

- **Module 6: Introduction to statistics (09 Hours)**
  Population and sample, parameter and statistics, descriptive statistics, graphical statistics.
- **Module 7: Statistical Inference and regression (15 Hours)**
  Parameter estimation, confidence, intervals, standard deviation, Bayesian estimation and hypothesis testing. Least squares estimation, analysis of variance, prediction, multivariate regression. Practical projects in R statistics or Python packages such as Numpy.

(d) Mode of Delivery
The study mode shall be blended learning including a combination of learner centered and teacher centered. The teacher shall give overview of the module and also provide examples illustrating the concept. The learner then shall do extra reading in the items in the module as well as doing the extensive module assignment and projects using statistical computing packages available for free in R, Python etc.

(g) Mode of Assessment
Assessment shall be by

1. At least one test and at least three assignments (40%)
2. Final examination (60%)

(h) Reading List

[1] Prasanna Sahoo Probability and Mathematical Statistical University of Louisville Louisville, KY 40292 USA, 2013:
(Springer Textbook) 1st Edition by

[2] Yakov G. Sinai and D. Haughton . Probability Theory: An Introductory Course 2012.

[3] Introduction to Probability and Statistics for Engineers and Scientists 5th Edition, Academic Press, 2014.

[4] Think Stats: Exploratory Data Analysis in Python, Second Edition, O'Reilly Media, 2014.

5.2.3 Software Development Project

Course Name:	Software Development Project
Course Code:	CSC 1202
Course Credit:	4
Contact Hours:	60
Year of Study:	1
Semester:	2

(a) Course Description
This course aims to give students an experience of developing complete software applications and systems. Focus will be put on practicing programming and program documentations combining all the skills and knowledge acquired in other courses so far.

(b) Course Objectives
The objectives of this course are to:

- practically give students skills of integrating different concepts of programming into a single application.
- introduce the student to hands on aspects of software development.
- nurture the students ability to independently read different sources of literature so as to identify what (s)he can use to develop his/her application.

(c) Learning Outcomes
By the end of the course unit, the student should be able to:

- Apply the skills he/she has acquired to integrate different concepts of programming into a single application;
- Apply the skills he/she has acquired to software development;
- Solve real life problems and challenges

(e) Detailed Course Content
The main content in the course is translation of a real life problem into a working computer program. For each year the course lecturer will provide a theme where students can be challenged to develop a software application that a real life problem. In addition, challenging practical challenges will be sourced from the industry and organisations that students can work on.

(f) Study Materials
The study materials will include examples and case studies of real-world challenges; software development tools and hardware equipment; open sources repositories on Github.

(d) Mode of Delivery
Learning will be largely by self - study/research. Students will be given programming Assignments that will be submitted after a specific period of time. The skills which that will be expected in the assignment will be indicated to the student. A member of staff will meet students at the issue of each of the assignment and address any outstanding issues as well as expectations. A minimum of six assignments will be given. Where necessary a guest speaker or project owner from the industry will be invited to provide additional context on the project or the state-of-the-art industry tools and methods.

(g) Mode of Assessment
Two assignments (each taking two weeks) and three assignments (each taking 3 weeks) will be given. The first two assignments will constitute the coursework (40%) while the last three assignments will constitute the examination mark (60%).

(h) Reading List
Students can read any literature like books and online tutorials that can help in addressing the problem in the project at hand. Students can also review available resources on Github and other common software repositories.

5.2.4 Operating Systems

Course Name:	Operating Systems
Course Code:	**CSC 1200**
Course Credit:	4
Contact Hours:	60
Year of Study:	1
Semester:	2

(a) Course Description
Operating systems are central to computing activities. There are two primary aims of an operating systems, i.e., to manage resources (e.g. CPU time, memory) and to control users and software. This class introduces the basic facilities

provided in modern operating systems. The course divides into three major sections. The first part of the course discusses concurrency: how to manage multiple tasks that execute at the same time and share resources. Topics in this section include processes and threads, context switching, synchronization, scheduling, and deadlock. The second part of the course addresses the problem of memory management; it will cover topics such as linking, dynamic memory allocation, dynamic address translation, virtual memory, and demand paging. The third major part of the course concerns file systems, including topics such as storage devices, disk management and scheduling, directories, protection, and crash recovery. After these three major topics, the class will conclude with a few smaller topics such as virtual machines.

(b) Course Objectives
The objectives of this course are to educate the students in:

- To introduce students to concurrency where students will learn how to manage multiple tasks that execute at the same time and share resources
- To introduce students to the problem of memory management
- To introduce students to file systems and issues of storage management.

(c) Learning Outcomes
By the end of the semester, students will:

- Explain the role of the operating system as a high level interface to the hardware.
- Use OS as a resource manager that supports multiprogramming
- Explain the implementation of CPU dispatch.
- Explain the implementation of memory management.
- Explain the performance trade-offs inherent in OS implementation

(e) Detailed Course Content

- **Module 1: Introduction to Operating Systems (4 CH)**
  Evolution of operating systems, The future of operating systems, Characteristics of current OSes, Categories of operating systems functionalities.
- **Module 2: Processes, Threads and Dispatching, Concurrent Threads (8 CH)**
  Threads and processes, dispatching, Process creation, Independent and cooperating threads.

- **Module 3: Locks and Condition Variables (8 CH)**
  Discuss basic lock operations, condition variables and monitors
- **Module 4: CPU Scheduling (4 CH)**
  Discuss simple scheduling alogorithms: FCFS, STCF. Priority-based scheduling, multiprocessor scheduling: processor affinity, gang scheduling.
- **Module 5: Implementing Locks and Deadlocks (4 CH)**
  Introduction to deadlocks and lock implementation.
- **Module 6: Linkers and Dynamic Linking (4 CH)**
  Introduction to linkers and dynamic linking.
- **Module 7: Dynamic Storage Management (4 CH)**
  Stack allocation, heap allocation, storage reclamation.
- **Module 8: Memory Management, Virtual Memory, Flash Memory (8 CH)**
  Dynamic memory relocation, base and bound relocation, multiple segments, paging, types of flash memory
- **Module 9: Storage Devices (4 CH)**
  Magnetic disk (hard disk), Reel-to-Reel magnetic tape, communicating with I/O devices: device registers, programmed I/O, interrupts, direct memory access.
- **Module 10: File Systems (8 CH)**
  File access patterns, file descriptors, Block cache, free space management, block sizes, disk scheduling.
- **Module 11: Virtual Machines (4 CH)**
  Introduction to virtual machines, history and usage of virtual machines, implementing virtual machine monitors,

(d) Mode of Delivery
The teaching pattern is by blended learning including lectures, tutorials, practical lab work, group discussion and class presentations.

(g) Mode of Assessment
The assessment will be done by mini-projects, tests and assignments (40%) and final examination (60%)

(h) Reading List

[1] Operating systems: Principles and Practice, Anderson, Thomas and Dahlin, Michael, 2nd Edition, Recursive books, 2014

[2] Operating System Concepts Silberschatz, A., Galvin P.B. and Gagne, G., 9th Ed, Wiley Publishing, 2012

5.2.5 Systems Analysis and Design

Course Name:	Systems Analysis and Design
Course Code:	IST 1204
Course Credit:	4
Contact Hours:	60
Year of Study:	2
Semester:	1

(a) Course Description
This course presents the processes, methods, techniques and tools that organizations use to build new information systems for their businesses, with particular focus on how computer-based technologies can most effectively contribute to the way business is organized. The course presents a systematic methodology for analyzing a business problem or opportunity, determining what role, if any, computer based technologies can play in addressing the business need. It articulates business requirements for the technology solution, specifies alternative approaches to acquiring the technology capabilities needed to address business requirements, and specifying the requirements for the information systems solution in particular. In- house development, development from third-party providers, or purchased commercial-off-the-shelf (COTS) packages are presented.

(b) Course Objectives

- Introduce students to the different contemporary approaches used for the analysis and design of information systems
- Equip students with the skills for initiating, specifying, and prioritizing information systems and Information Technology projects and for determining various levels of feasibility for these projects
- Equip students with the different techniques commonly used to collect, analyze and document business requirements, in order to cause productive change in a way a business operates

- Equip students with the different methodologies for designing and communicating a business solution to a problem using formal techniques, and contemporary case tools in process and data modeling
- Equip students with the skills needed to manage information systems projects using formal project management methods.
- Introduce students to various systems acquisition alternatives, including packaged systems, and outsourced design and development resources.

(c) Learning Outcomes
On successful completion of this course students will be able to.

- Explain the different approaches used for the analysis and design of information systems
- Demonstrate understanding of the initiation, specification and prioritization of information systems and Information Technology projects and there different levels of feasibility
- Use contemporary techniques to collect, analyze and document business requirements, in order to cause productive change in a way a business operates
- Use contemporary techniques for designing and communicating a business solution to a problem using formal techniques, and contemporary case tools for data modeling
- Demonstrate understanding of the managing information systems projects using formal project management methods.
- Explain the various systems acquisition alternatives, including packaged systems, and outsourced design and development resources

(e) Detailed Course Content

- Introduction to Systems Analysis and Design - **12 hours**
  - ∗ The different Analysis & Design Philosophies and Approaches for Information Systems: structured SDLC, UML, agile methods, soft systems methodology
  - ∗ Managing Systems Projects.
  - ∗ Configuration and Change Management
- Planning IT based projects - **4 hours**
  - ∗ Analysis of project feasibility

    - ∗ Identification of opportunities for IT-enabled organizational change
- – Analysis and Specification of System Requirements - **12 hours**
    - ∗ Analyzing of business requirements
    - ∗ Requirements Gathering
    - ∗ Requirements Modelling
    - ∗ Data and Process Modelling
- – High-level System Design of Information systems - **12 hours**
    - ∗ User Interface Design
    - ∗ Data Design
    - ∗ System Architecture
- – System Deployment and Implementation - **8 hours**
- – System Verification and Validation - **8 hours**
- – Managing Systems Support and Security - **4 hours**

(d) Mode of Delivery
Blended learning including

- – Lectures
- – Practicals using CASE Tools
- – Online learning management systems
- – Class discussions and presentations
- – Group project-based assignments

(g) Mode of Assessment
Assessment will be by continuous assessment through practical exercises and Coursework (40%) and final exam (60%)

(h) Reading List

[1] Systems analysis and design Shelly, Gary B and Rosenblatt, Harry J, Cengage Learning, 2011

[2] Knowledge management: organizing knowledge based enterprises Hawryszkiewycz, Igor Macmillan International Higher Education, 2009

[3] Information systems analysis and design Wang, Shouhong and Wang, Hai Universal-Publishers, 2012

5.3 Year I Recess Term

5.3.1 Practical Skills Development

Course Name:	Practical Skills Development
Course Code:	CSC 1304
Course Credit:	5
Contact Hours:	75
Year of Study:	1
Semester:	Recess Term

(a) Course Description
This is a practical course designed to equip students with essential practical skills in a selected Computer Science area. The courses builds upon the foundation knowledge gained in the first and second semesters. The exact area of focus will be determined by the Course Lecturer. The students are to be supervised by staff. The practical sessions can be run in collaboration with the industry or government organisations dependent on the topic. For example, for teaching of student to integrate mobile money payment services in software systems can be done in collaboration with the telecom companies.

(b) Course Objectives
The aims of the course are to provide the student with:

- ability to study a real word challenge and formulate a relevant digital solution
- Knowledge and techniques on how to implement a working hardware orand software system
- Knowledge in project planning and development, team management, organization as applied to a complete system development.
- Skills to document and present a digital product to potential users.

(c) Learning Outcomes
Students who successfully complete this course of study will be able to:

- understanding of the process involved in the translation of a real world problem into a digital solution.
- demonstrate ability to implementing a working digital solution.
- document and present a digital product to the target audience.

- work as part of computer science project execution team.
- demonstrate an understanding of the tools for implementation of a working digital solution.

(e) Detailed Course Content
The content of course is not fixed. The project will be determined by the current industry needs and challenges. The broad themes of practical skills development include but not limited to:

- Mobile applications development
- Application and use RESTful APIs for example use of Mobile Money open API.
- Internet of Things and Embedded systems applications
- Robotics applications
- Block chain applications
- Implementation of projects
- Network and system administration
- Hardware maintenance
- Computer assembly. This can be done within the School of Computing and Informatics Technology or any other unit in Makerere University. Students will write a report at the end of the course.

(f) Study Materials
The course materials will be dependent on the selected theme. These will include online resources and text books as determined by the supervisor.

(d) Mode of Delivery
The mode study will be blended learning including tutorials, self-study research and development, group and team discussions.

(g) Mode of Assessment
The assessment will be done by technical project documents, progress review meetings (40%) and project report, poster presentation, project demo and presentation (60%).

(h) Reading List
The reading materials will vary dependent on the selected thematic area and project.

5.4 Year II Semester I

5.4.1 Computer Networks

Course Name:	Computer Networks
Course Code:	BSE 2106
Course Credit:	4
Contact Hours:	60
Year of Study:	2
Semester:	1

(a) Course Description

This course examines principles, design, implementation, and performance of computer networks and data communication. The aim is to sharpen students understanding and skills in computer networking and data communication. Subjects for discussion include: Internet protocols and routing, local area networks, wide area networking, wireless communications and networking, performance analysis, congestion control, TCP, network address translation, multimedia over IP, switching and routing, mobile IP, peer-to-peer networking, network security, and other current research topics and technologies

(b) Course Objectives

The aims of the course are;

- To provide a solid basis on the theoretical and practical understanding of computer networks.
- To introduce students to standards and guidelines in computer and data communication networks.
- To impact knowledge and skills relevant for the design, implementation and maintenance of modern computer communication networks.
- To introduce students to emerging technologies in computer networks

(c) Learning Outcomes

Upon successful completion of this course, the student should be able to;

- Describe theoretical concepts of computer communication systems, including standards, protocols, and infrastructure devices e.g. routers and switches.

- Design, install and manage a simple Local Area Network or A campus Wide Area Network
- Describe emerging communication technologies particularly wireless and fiber optic technologies
- Deploy sound network security practices and tools
- Describe emerging research directions in computer communication networks
- To write a few network scripts.

(e) Detailed Course Content

- **Module 1: Computer Networks and the Internet (4 CH)** Introduction to computer networks and the internet, network edge and core, delay, loss and throughput in a packet, protocol layers and their service models.
- **Module 2: Application Layer (8 CH)**
  Network services and applications: DNS, HTTP, SMTP, FTP, peer-to-peer systems
- **Module 3: Transport Layer (8 CH)**
  Network transport architectures, Demultiplexing & Multiplexing, TCP, UDP, TCP congestion control and Reliable data transfer.
- **Module 4: Network Layer (8 CH)**
  Routing and forwarding, intra-domain, inter-domain routing algorithms, Mobile IP, Queuing and scheduling.
- **Module 5: The Link Layer and Multimedia Networking (12 CH)**
  Link layers and local area networks, Ethernet, WiFi, and mobility/Wireless and Mobile Networks. Multimedia communications and quality of service
- **Module 6: Network Security and Management (12 CH)**
  Network measurement, inference, and management, Network security (ACL, IPSec, SSL etc), Network programming, Network experimentation and performance analysis and Protocol verification.
- **Module 7: Emerging Trends (8 CH)**
  Emerging trends in computer networks e.g Internet of Things, IPV6, 5G etc

(d) Mode of Delivery
The teaching pattern is by blended learning including lectures, tutorials, practical lab work, group discussion and class presentations

(g) Mode of Assessment
Assessment will be in terms of tests and Assignment (40%) and final examination (60%).

(h) Reading List

[1] James F. Kurose, Keith W. Ross, Computer Networking A Top-Down Approach, sixth Edition, Pearson Education, 2013.

[2] Larry L. Peterson, Bruce S. Davie, Computer Networks: A Systems Approach, Fifth Edition, Morgan Kaufmann Publishers, 2011.

5.4.2 Database Management Systems

Course Name:	Database Management Systems
Course Code:	CSC 2107
Course Credit:	4
Contact Hours:	60
Year of Study:	2
Semester:	1

(a) Course Description
This course trains the student in theoretical and practical aspects of modeling and developing database applications. The course covers the rationale for databases in the current systems development environment, the modeling, implementation and testing/tuning database driven applications. It also covers advanced topics tile indexing hashing, security and performance.

(b) Course Objectives
The objectives of the course are to equip the student with knowledge to

1. Carry out system and data requirement analysis for database driven applications
2. Model, map and tune database schema
3. Implement typical database driven applications
4. Monitor and tune operational database systems
5. Handle some security and performance issues related to databases

(c) Learning Outcomes
By the end of the course, the learners should be able to

1. Develop a system and data requirements document for an application
2. Carry out conceptual, logical and physical modeling of a typical database application
3. Implement a database driven application
4. Diagnose and fix performance related problems for operational systems
5. Explain security, performance and integrity issues in database systems

(e) Detailed Course Content

- **The Database Application Development Life-Cycle**(15CH)
  The history of databases, basic definitions in DBMS, file based systems and database based systems, data models, players in the database field, changes in database applications trends. Description of the the DB application process, decisions to be made and their merits/demerits, common mistakes made in the development process.
- **Database Applications Modeling** (15 CH)
  System and data modeling process, documentation standards, The data modeling process, Conceptual modeling, Logical modeling, Physical modeling, enhanced conceptual modeling, normalisation and denormalisation, tuning operational systems, Data dictionary development.
- **SQL, Scripting and Mini Project** (20 CH)
  SQL statements, PL SQL, survey of a scripting language, Students will then be given a micro project in which they will develop and document the development of a small database driven application.
- **Advanced topics in DBMS**(10 CH)
  Relational algebra, query processing, Indexing, Hashing, emerging architectures (parallel databases, distributed databases)

(f) Study Materials
The materials shall include textbooks, laboratories for demonstration, online resources and case studies

(d) Mode of Delivery
The study will be blended learning including a mix of student centered and learning teacher centered pedagogy. Specifically the student is expected to do some guided reading as well as practically implementing a database application.

(g) Mode of Assessment
Assessment will include

1. At least one assignment per module and atleast one test that will constitute progressive assessment (40%)
2. Final examination that will constitute 60%

(h) Reading List

[1] Thomas Conolly and Carollyn Begg(2010) Database systems. A practical approach to design, implementation and management (Addison-Wesley)

[2] Ramez Elmasri, and Shamkant B. Navathe,(2016) Fundamentals of Database Systems, (7th Edition)

5.4.3 Embedded and Real-time Systems

Course Name:	Embedded and Real-time Systems
Course Code:	CSC 2118
Course Credit:	4
Contact Hours:	60
Year of Study:	1
Semester:	1

(a) Course Description
Embedded and real-time systems are becoming an integral part of modern Computer Science applications. Most computers today's come in the form of embedded and real-time systems in cars, refrigerators, coffee machines, security systems, etc. This courses aims to equip students with the understanding of embedded and real-time systems and skills to design, analyse and develop practical applications of embedded systems with real-time constraints in the areas of agriculture, health, entertainment, security, traffic control and management, automotive systems, energy, environment, robotics, smart cities, and others. The course topics include

(b) Course Objectives
The objectives of the course are:

- ensure that students gain an understanding of the principles, concepts and technologies for embedded and real-time systems

- ensure that students understand the programming, operating systems and software tools for developing software for embedded and real-time systems
- provide students with the skills to design and develop embedded systems
- introduce students to basic electronics, different types of micro-controllers and common architectures.

(c) Learning Outcomes
By the end of the course unit, the student should be able to:

- design and develop embedded systems that address a practical challenge
- demonstrate an understanding of the software tools used for modern embedded systems development
- be able to build embedded systems with real-time constraints.
- demonstrate an understanding of micro-controllers and common architectures. capabilities.

(e) Detailed Course Content

- **Module 1 Introduction to Embedded and Real time systems (10 Hours):** Properties of embedded systems, design, practical applications. Embedded systems architecture. Basic electronics and circuit for embedded systems.
- Module 2: Micro-controllers and Microprocessors (10 Hours). Basic structure of a Micro-controller. Examples of micro-controllers (ATTiny85, PICx, 32-Bit) and Development Boards.
- Module 3: Embedded systems with Arduino (15 Hours). Introduction to Arduino boards, The Arduino IDE, Arduino Board models, Shields, Arduino programming and Arduino C Library, Arduino example projects.
- Module 4: I/O Interfaces (10 Hours). Switches, analog inputs, high-power digital outputs, sound interfaces, serial interfaces, actuators, sensors, level conversation and LED display interfaces.
- Module 5: Schematic design and simulation programs (09 Hours). Circuit schematics and simulator programs, PCB designs.
- Module 6: Real-time Scheduling and Real-time Operating Systems (06 Hours). Scheduling, priorities, re-entrancy, programming multi-core based embedded systems.

(f) Study Materials
Tex-books, open source micro controllers such as Arduino boards, and software development tool kits.

(d) Mode of Delivery
This course will delivered by blended learning including class lectures and practical sessions. Students will be introduced real life examples and illustrations to enable them apply the acquired knowledge with the business context. Students will be required to analyse business domain problem and come up with communications to address the needs as part of their coursework assessment

(g) Mode of Assessment
Course work assessment will contribute (40%) and examination will contribute (60%).

(h) Reading List

[1] E. A. Lee and S. A. Seshia, Introduction to Embedded Systems - A Cyber-Physical Systems Approach, LeeSeshia.org, 2011.

[2] Elecia White Making Embedded Systems, O'Reilly Media, Inc., 2011

[3] Paul Scherz and Simon Monk Practical Electronics for Inventors 4th Edition, Mc Graw Hill Education, 2016.

[4] Heath, Steve. Embedded systems design, *Oxford: Newnes*, 2003

[5] Wolf, Marilyn. TComputers as components: principles of embedded computing system design, Cambridge, MA: Morgan Kaufmann, 2017.

[6] dawsonera. Computers as components: principles of embedded computing system design, *London: Elsevier/Morgan Kaufmann.s*, 2008

5.4.4 Artificial Intelligence

Course Name:	Artificial Intelligence
Course Code:	CSC 2114
Course Credit:	4
Contact Hours:	60
Year of Study:	2
Semester:	1

(a) Course Description

Artificial intelligence (AI) is a research field that studies how to encode the intelligent human behaviors onto a computer. The ultimate goal of AI is to make a computer that can learn, plan, and solve problems autonomously. Artificial Intelligence has been studied for of over half a century but there is not computer has been made that is as intelligent as a human being in all aspects. However, there are successful applications such as computers playing chess, manufacturing and in some cases, a computer equipped with artificial intelligence technology can be even more intelligent than humans. The Deep Blue system that defeated the world chess champion is a well-know example.

The course focuses on the theory and practice of Artificial Intelligence. It will concentrate on the study of modern techniques for computers to represent task-relevant information and make intelligent decisions towards the achievement of goals. The search and problem solving methods are applicable throughout a large range of industrial, civil, medical, financial, robotic, and information systems. The course investigates questions about AI systems such as: how to represent knowledge, how to effectively generate appropriate sequences of actions and how to search among alternatives to find optimal or near-optimal solutions. We will also explore how to deal with uncertainty in the world, how to learn from experience, and how to learn decision rules from data.

(b) Course Objectives

- To provide students with comprehensive and in-depth knowledge of AI principles and techniques by introducing AIs fundamental problems, and the state-of-the-art models and algorithms used to undertake these problems.
- To expose students to the frontiers of AI-intensive computing while providing a sufficiently strong foundation to encourage further research.

(c) Learning Outcomes
On successful completion of this course students will be able to.

- Explain what constitutes "Artificial" Intelligence and how to identify systems with Artificial Intelligence.
- Compare AI with human intelligence, and discuss its strengths and limitations and its application to complex and human-centered problems.
- Discuss the core concepts and algorithms of AI, including intelligent agents; search algorithms; adversarial search; Constraint satisfaction problems; Utility theory, Uncertainty and probability theory, probabilistic reasoning in AI; Bayesian networks, Nearest neighbour methods;clustering.
- Apply the basic principles, models, and algorithms of AI to recognize, model, and solve real world problems that require intelligence.
- Analyze the structures and algorithms of a selection of techniques related to searching, reasoning, machine learning, language processing, computer vision, image processing and robotics.

(e) Detailed Course Content

- **Module 1: Introduction to Artificial intelligence (AI). 4 hours**
  The module provides a broad introduction to AI. We discuss what Artificial intelligence is and its Applications We also discuss intelligent agents, Environment types and types of intelligent Agents.
- **Module 2: Planning- uninformed search. 8 hours**
  The module discusses intelligent agents that plan. We discuss the methods used to find a plan for solving a problem using only problem definition information. We will discuss Depth first search, Breadth first search, iterative deepening search, and uniform cost search strategies for finding a

solution in such situations. We will also discuss the advantage and disadvantages of using each of the strategies

- **Module 3: Planning - A^* search and Heuristics search. 8 hours**
  We discuss methods used to find a plan for solving when we have some prior information to guide the plan. We will discuss what a heuristic is, space graph search, A^* search, greedy search, optimally of A^* graph search, admissible and consistent heuristics, generating admissible and consistent heuristics. We will also discuss how to generate the heuristics.
- **Module 4 - Constraint Satisfaction Problems (CSPs) (12 hours)**
  The module introduces a way of generating plans that satisfy some constraints. We discuss the basic concepts of a constraint satisfaction problems: What is a CSP? Types of CSPs, Constraint graph, Applications of CSPs, CSP formulation and solving a CSP by Backtracking search. We will also discusses various ways of speeding up the backtracking search: filtering, Arc Consistency, ordering, exploring the constraint graph structure, Iterative Algorithms for CSPs such as minimum conflict algorithm.
- **Module 5: Adversarial Search (8 hours)**
  We discuss a way of finding a plan where one or more agents a participating. We will discuss game playing, minmax algorithm, Alpha-Beta Pruning technique, Uncertainty and Utilities, Expectimax Search, utilities, multiagent utilities and Maximum expected utility (MEU).
- **Module 6: Non-Deterministic Search and Reinforcement learning. 10 hours**
  The modules provides an introduction to Markov Decision Processes and Policies as well as an introduction Reinforcement learning.
- **Module 7: Bayesian Network (10 hours)**
  The module introduces the Bayesian network, Representation , Independence and Inference

(d) Mode of Delivery
Teaching and learning is implemented through blended learning including Lectures, Tutorials sessions and Assignments/Quizzes. Lectures will introduce and motivate the basic concepts of each topic. Significant discussions and two-way communication are also expected during lectures to enrich the learning experience. Tutorials provide opportunities for obtaining feedback. The assignments/quizzes willreinforce theoretical concepts by their application to problem solving. Assignments will be done via programming work using Python programming language. Students will be expected to make presentations of their assignments for discussion in class.

(g) Mode of Assessment
Assessment will be by continuous assessment through practical exercises and Coursework (40%) and final exam (60%)

(h) Reading List

[1] Russell Stuart Jonathan and Norvig Peter. Artificial Intelligence: A Modern Approach. 4^{th} Edition, Prentice Hall, 2021.

[2] Philip C Jackson. Introduction to Artificial Intelligence. *Second enlarged Edition.*

5.4.5 Discrete Mathematics

Course Name:	Discrete Mathematics
Course Code:	CSC 2105
Course Credit:	3
Contact Hours:	45
Year of Study:	2
Semester:	1

(a) Course Description

The course applies mathematics to finite or discontinuous quantities in order to master the process of problem-solving, communication, reasoning, and modeling. It gives a basic understanding of mathematical structures that are fundamentally discrete. In particular, this course is meant to introduce logic, proofs, sets, relations, functions, counting, and probability, with an emphasis on applications in computer science. Concepts and notations from discrete mathematics are useful in studying and describing objects and problems in computer algorithms and programming languages.

(b) Course Objectives

At the end of this course, students should demonstrate the ability to:

- use mathematical logic to solve problems
- use iterative and recursive processes to prove properties of integers
- use sets to codify mathematical objects

- use discrete functions and relations to solve problems
- use a problem-solving approach in applying counting techniques in order to determine probabilities

(c) Learning Outcomes

At the end of this course, students should:

- use correctly the classical notions of logic : implications, equivalence, negation, proof by contradiction, proof by induction, quatificators
- use set theory: union, intersection, complementary, maps, bijection, injection, surjection
- know the main formulas in combinatorics: enumerations of subsets, enumerations of injections, surjections, bijections
- know the main definitions, some classical theorems on graphs and apply graphs in concrete situations

(e) Detailed Course Content

- Module 1 (6 hours) - **Logic**: Propositional and First Order Logic
- Module 2 (6 hours) - **Functions** and The Set Theory. Basic set operations, one-to-one, onto, inverses, composition and graphs.
- Module 3 (6 hours) - **Relations** and the Number Theory
- Module 4 (09 hours) - **Methods of Proof**: Direct Proofs, Implications, Contraposition, Contradiction, Proof by Cases, Induction
- Module 5 (06 hours) - **Counting**: Sum Rule, Product Rule, Permutations, Combinations, Combinatorial Proofs,
- Module 6 (3 hours) - **Recursion:** recursive definitions, recursive algorithms. Solving recurrences.
- Module 7 (9 Hours) - **Graphs and Trees:** Terminology, Representing Graphs, Isomorphism, Eulerian and Hamiltonian Graphs, Bipartite Graphs and Matchings, The Stable Marriage Problem, Graph Colorings.

(d) Mode of Delivery

Teaching and learning will be by blended learning including lectures, tutorials and practical sessions.

(g) Mode of Assessment

The assessment will be done by tests, take home assignments (40%), and written examination (60%)

(h) Reading List

1. Discrete Mathematics and Its Applications by Kenneth H. Rosen, 7th Edition, McGraw Hill, 2012
2. Discrete Mathematics, by Kenneth A. Ross and Charles R. B. Wright, Fourth Edition
3. Discrete and Combinational Mathematics: An Applied Introduction, by Ralph Grimaldi, 4th Edition
4. Discrete Mathematics with Applications by Susanna S. Epp

5.5 Year II Semester II

5.5.1 Research Methodology

Course Name:	Research Methodology
Course Code:	IST 2203
Course Credit:	4
Contact Hours:	60
Year of Study:	2
Semester:	2

(a) Course Description
Research is essential in nearly all aspects of life. This course unit enables students to learn and apply principles of conducting scientific research. It caters for the rationale of doing research, the research process, findings presentation as well as validation of findings. The course also prepares the students on the process of conducting the final year project.

(b) Course Objectives
To objectives of the course are to equip students with skills to:

- Formulate a research problem

- Document and justify the research aims
- Select appropriate methods of solving a research problem
- Correctly correct and analyze data
- Document and present findings

(c) Learning Outcomes
A Student who has undertaken this course unit will be able to learn skills that will enable him/her to successfully undertake a research project. He/she will be able to:

- Identify a relevant or significant research problem.
- Identify the aims of a research project that can solve a given problem.
- Select appropriate research methods to be used in solving a given project.
- Select appropriate data collection techniques that can be used to gather data required to solve a given project.
- Select appropriate data analysis techniques and use them to process collected data, interpret data analysis results.

(e) Detailed Course Content

- **Introductory concepts of research** :
  what is research, understanding the research process and fundamental concepts of research, how to formulate a research problem, research objectives, research questions, how to define scope, how to conduct literature review in a given study etc. **(12 Hours)**
- **Research methods:**
  Student learns the various quantitative and qualitative research methods. Student also learns the data gathering techniques. **(12 Hours)**
- **Research evaluation:**
  The evaluation phase of a research project, testing and validation of developed tool or research results. E.g. How to evaluate, testing, and validate research results; what are the evaluation criteria? What are the validation methods? What is one testing in the testing phase? **(12 Hours)**
- **Research application:**
  This covers the practical application of concepts learned from parts I to III. The lecturer suggests one or several class projects that require practical application of concepts learned from parts I to III.**(12 hours)**

- **Report writing:**
  How to prepare a research report, how to report research results/findings, how to present feedback or findings from evaluating the developed tool or research results. **(12 Hours)**

(f) Study Materials
Text books, conference and journal publications, and online resources.

(d) Mode of Delivery
Blended learning including:

- Lectures
- Online learning management systems
- Class discussions and presentations
- Group project-based assignments

(g) Mode of Assessment
The course will be assessed by Course work (individual assignments and tests) 40 % and Final examination 60 %

(h) Reading List

[1] Zikmund, W.G. Thomson South Western Business Research Methods (8th Ed.)(2010)

[2] Leedy, P.D., & Ormrod, J.Practical Research: Planning and Design Paperback (2004)

[3] Sounders, M., Lewis, P. & Thornhill, A. Research Methods for Students. 3rd edition(2003) UK, Financial times, Prentice hall

[4] Graziano, A.M, Michael, L. Research Methods: A process of Inquiry Raulin, Hardcover, Prentice Hall(2006)

5.5.2 Introduction to Machine Learning

Course Name:	Introduction to Machine Learning
Course Code:	CSC 2206
Course Credit:	4
Contact Hours:	60
Year of Study:	2
Semester:	2

(a) Course Description

Machine learning is the science of getting computers to act without being explicitly programmed. Over the past years, machine learning has given us self-driving cars, practical speech recognition, effective web search, and a vastly improved understanding of the human genome. Machine learning is being used in many applications now days and many researchers think it is the best way to make progress towards human-level artificial intelligence.

The machine learning course aims at introducing the fundamental concepts of machine learning. In the course, students learn about the most effective machine learning techniques, and gain practice in implementing them and making them work. More to that students will learn about the theoretical underpinnings of learning and practical know-how needed to quickly and effectively apply these techniques to new problems.

The course introduces models and algorithms for regression, classification and clustering. Students taking the course will need to be familiar with linear algebra, probability theory, and programming specifically in Python.

(b) Course Objectives

- To introduce methods for learning from data.
- To provide the necessary mathematical background to enable students to understand how the methods work.
- To provide an understanding on how to evaluate the performance a machine learning models and how to get the best performance from them.

(c) Learning Outcomes

On successful completion of this course students will be able to.

- develop an appreciation for what is involved in learning from data.
- understand a wide variety of learning algorithms.
- understand how to apply a variety of learning algorithms to data.
- understand how to perform evaluation of learning algorithms and model selection.

(e) Detailed Course Content

- **Module 1: Introduction to Machine Learning (6 CH)**
  This module introduces the students to the major concepts of machine learning. It introduces the broad concepts of learning. Concepts like: supervised learning, unsupervised learning, data representation, and overfitting will be discussed in this module.
- **Module 2: Machine Learning Basics (10 CH)**
  The module will introduce the basis concepts and distributions used in building machine learning models. In the module we discuss the Bayesian Theorem, Gaussian Distribution, Gaussian Mixture Models and Maximum Likelihood Estimation technique.
- **Module 3: Feature selection and Evaluation metrics (4 CH)**
  In this module we discuss methods of feature selection and the methods for evaluating the models.
- **Module 4: Supervised learning Models (20 CH)**
  The module introduces various supervised learning models in detail. In the module we discuss Linear Regression, Logistic Regression, Naive Bayes Classifier, Decision Tree, Support Vector Machine (SVM), and K-nearest Neighbour classifier. We also discuss Regularisation and Gradient Descent.
- **Module 5: Unsupervised Learning (8 CH)**
  This module introduces the idea of learning from data by clustering. We discus the K-Means and Expectation Maximisation methods.
- **Module 6: Introduction to Deep Learning (12 CH)**
  The module provides an introduction to Deep learning. Concepts concerning Multilayer Perceptron, Convolutional Neural Network and Recurrent Neural Network will be discussed.

(f) Study Materials
Textbooks, Assignments, online materials. The programming and program demonstrations will be offered using the Python programming language. Students should possess a laptop computer for doing in-class programming assessments and demonstrations.

(d) Mode of Delivery

- Teaching and learning will employ blended approaches implemented through online or face to face Lectures, Tutorials sessions and Quizzes. Lectures will introduce and motivate the basic concepts of each topic. Significant discussions and two-way communication are also expected during lectures to enrich the learning experience. Tutorials provide opportunities for obtaining feedback. Quizzes will reinforce theoretical concepts by their application to problem solving.

– Home works and Assignments: The best way to learn about a machine learning method is to program it yourself and experiment with it. So the assignments will generally involve implementing machine learning algorithms, and experimentation to test your algorithms on some data. The students may asked to summarize their work, and analyze the results, in brief (3-4 page) write ups. The implementations will be done in Python.

(g) Mode of Assessment
Assessment will be by continuous assessment through programming assignments (20%) , written assignments and quizzes (20%) and Final exam (Practical exam (30) and written (30%)).

(h) Reading List

[1] Bishop, C., M. Pattern Recognition and Machine Learning. Springer 2006

[2] Tom Mitchell. Machine Learning McGraw Hill.

[3] Andreas Mueller. Scikit-learn Tutorial: Statistical-Learning for Scientific Data Processing.

[4] Shai Shalev-Shwartz and Shai Ben-David Understanding Machine Learning: From Theory to Algorithms.

[5] Stephen Marsland. Machine Learning: An Algorithmic Perspective.

[6] Willi Richert and Luis Pedro Coelho. Building Machine Learning Systems with Python

5.5.3 Robotics

Course Name:	Robotics
Course Code:	CSC 2207
Course Credit:	4
Contact Hours:	60
Year of Study:	2
Semester:	11

(a) Course Description
Robotics as an application combines many fields and allows automation of products such as cars, vacuum cleaners, and factories. The course aims at equipping the students with the fundamental concepts used in automating robotics; specifically the students will be introduced to the basic computation concepts used in robotics. The focus will be mainly on mobile robotics, and will cover the basic issues in this dynamic field via lectures and a large practical element where students work in groups.

Topics include will Introduction to Robotics, Locomotion Kinematics and planning, perception and sensing, Planing, localization and navigation. The mathematical basis of each area will be emphasized, and concepts will be motivated by using common robotics applications and programming.

(b) Course Objectives

- To understand the basic concepts and theory governing the programming of robots that perform autonomous tasks such as navigation and manipulation.

(c) Learning Outcomes
On successful completion of this course students will be able to.

- Understand the defining properties of a robot: sensing and action, linked by processing
- Understand an overview of the practical issues of modern-day mobile robotics.
- Understand Robot locomotion methods, particularly wheel configurations and uncertainty in motions.
- Understand the use of simple sensors in reactive, behavioural programming.
- Understand and implement localization and mapping algorithms using different sensor modalities.
- Understand the concepts of different approaches for motion planning such as roadmaps, feedback control and sampling based methods.
- Apply the tools learned in the class to physical robots.

(e) Detailed Course Content

- **Module 1: Introduction to Robotics (4 CH)**
  What is robotics, applications and challenges in robotics, categories of robotics, robots and types, mobiles robots: state of the art.
- **Module 2: Locomotion and Kinematics (8 CH, 20 PH)**
  Basic 3D geometry and notation, Introduction kinematics, Open and Closed Kinematics, Coordinate systems and relationship, Homogeneous Transforms, Direct Vs inverse kinematics, Kinematics- dynamics and control, the robotics locomotion scheme, locomotion concepts, characteristics of locomotion, legged locomotion, locomotion of wheel, Standard Wheel Configuration, locomotion Notion, mobile robot locomotion, Mapping Wheel rotation Speed to Velocity, motion and State on a 2D Plane,Position-Based Path Planning,
- **Module 3: Perception and Sensing (9 CH, 20 PH)**
  What is perception? what are the challenges to perceptions? mobile robot control system, sensors for mobile robot, perception for Mobile robots and it's challenges, evolution of robotic sensors, classification of sensors, sensor measurements, sensor outlines: single and multiple value sensors, Servoing, combining: sensing/ Action Loops, combining Sensors: World Model Approach.
- **Module 4: Localization and Navigation (9 CH, 20 PH)**
  What is localization?, Challenges to localization, Localization-based navigation versus programmed solutions, Belief representation, Map representation, Probabilistic Map-Based Localization, Other Example s of Localization Systems, Autonomous Map Building.

(f) Study Materials
Textbooks, Research paper, Assignments, projects, online materials, e-learning

(d) Mode of Delivery
Blended learning including:

- Teaching and Learning is implemented through online or face to face lectures, tutorial sessions, Assignments/Quizzes, Research papers, group discussion and class presentations. Lectures will introduce and motivate the basic concepts of each topic.
- Significant discussions and two-way communication are also expected during lectures to enrich the learning experience. Tutorials provide opportunities for obtaining feedback. The assignments/quizzes will reinforce theoretical concepts by their application to problem solving

- Project programming work: The best way to learn about a Robotics is to program it yourself and experiment with it. The project implementation will using the Raspberry Pi boards computer and Python using BrickPi interface boards and Lego NXT motors and sensors.

(g) Mode of Assessment
Assessment will be by continuous assessment (40%) distributed as follows: Assignments/Quizzes and group research (10%), Tests (10% and projects work (20%)) and Final exam (60%) distributed as follows Written (40%) and practical exam (20%).

(h) Reading List

[1] Robin R. Murphy. Introduction to AI Robotics MIT Press, 2000.

[2] Maja J. Mataric. The Robotics Primer MIT Press, 2007.

[3] Steven M. LaValle. Planning Algorithms. Cambridge University Press, 2006.

[4] John J. Craig. Introduction to Robotics. Mechanics and Control. Third Edition, 2005.

[5] Kevin M. Lynch and Frank C. Park Modern Robotics. Mechanics, Planning and Control. 2017.

[6] Mark W. Spong and Seth Hutchinson, and M. Vidyasagar Robot Modeling and Control. First Edition.

5.5.4 Software Quality and Verification

Course Name:	Software Quality and Verification
Course Code:	CSC 2208
Course Credit:	4
Contact Hours:	60
Year of Study:	2
Semester:	2

(a) Course Description
This is a course that puts emphasis on the concepts and practices that reduce software cost while increasing reliability and modifiability. The quality of software systems developed is a major concern in today's world of automation.

This course focuses on how to build quality in both the software product and process. Software verification ensures that the product is being built according to the requirements and design specifications.

(b) Course Objectives
The objectives of this course are to provide the students with:

- Knowledge of factors that affect the quality of a software product.
- The techniques to produce high-quality and reliable software, regardless of the software system's level of complexity.

(c) Learning Outcomes
Students who successfully complete this course of study will be able to:

- Understand software testing and quality assurance as a fundamental component of the software life cycle.
- Define the scope of software quality assurance and testing projects.
- Efficiently perform quality assurance and testing activities using modern software tools.
- Describe the components of configuration management.

(e) Detailed Course Content

- **Module 1: Introduction (6 CH)** what is software quality and its importance; Software Quality characteristics;General Quality views; User versus producer expectations; Quality frameworks, standards and models; Software quality management and documentation; Quality Dimensions, The Software Quality Dilemma, Achieving Software Quality
- **Module 2: Software Quality Assurance (8 CH)** Components; Correctness and Defects; Defects and Quality; Defect prevention, reduction and containment; QA in software development and maintenance processes; Verification and Validation concepts;
- **Module 3: Quality Engineering (6 CH)** activities and processes; Quality engineering in software processes; context and cost. Software Quality Control and why it is needed;
- **Module 4: Software Quality metrics (8 CH)** purpose; types of software audits; Software Quality Indicators; How can we measure software quality?

- **Module 5: Software Testing (12 CH)** the software testing process, planning and testing strategies, Test management and automation; Test case design; Alpha and Beta Testing; testing and quality assurance; functional versus structural testing; Testing and the Software Life Cycle; Static testing: techniques and types; Dynamic testing
- **Module 6: Software configuration management (8 CH)** items and management; tasks and organization; baseline and intermediate software configuration versions;
- **Module 7: Formal Verification (12 CH)** Basic Concepts: Formal Verification and Formal Specification; Axioms; pre/post conditions; Weakest pre-conditions; Software inspection and walkthroughs - informal code reviews, formal inspection;

(d) Mode of Delivery
The teaching pattern is by blended learning including including lectures and tutorials, practical lab work, group discussions and class presentations.

(g) Mode of Assessment
The assessment will be done by tests and assignments (40%) and final examination (60%)

(h) Reading List

[1] Stephen H. Kan. Metrics and Models in Software Quality Engineering (2nd Edition), 2001.

[2] Gopalaswamy Ramesh and Srinivasan Desikan. Software Testing: Principles and Practices, 2007.

[3] Tom Badgett, Corey Sandler, Glenford J. Myers. The Art of Software Testing, 3rd Edition, 2011.

[4] Jeff Tian. Software Quality Engineering, 2005.

5.5.5 Cloud Computing

Course Name:	Cloud Computing
Course Code:	CSC 2209
Course Credit:	4
Contact Hours:	60
Year of Study:	3
Semester:	2

(a) Course Description
Cloud computing is a computing paradigm, where a large pool of systems are connected in private or public networks, to provide dynamically scalable infrastructure for application, data and file storage. With the advent of this technology, the cost of computation, application hosting, content storage and delivery is reduced significantly. This course gives an introduction to cloud computing technologies, applications as well emerging paradigms such as Cloud Native Computing.

(b) Course Objectives
The objectives of this course are to provide the students with:

- Knowledge of cloud computing models and services.
- An understanding of public and private clouds.
- Techniques to set up, manage and protect data in the cloud.
- An understanding of emerging cloud native paradigm and DevOps concepts, methodologies and tools.

(c) Learning Outcomes
By the end of the course, students should be able to:

- Analyze the trade-offs between deploying applications in the cloud and over the local infrastructure.
- Compare the advantages and disadvantages of various cloud computing platforms.
- Deploy applications over cloud computing infrastructures such as Amazon Web Services, Windows Azure, and Google AppEngine.
- Identify security and privacy issues in cloud computing.
- Work with modern cloud native and DevOps tools such as Kubernetes and Docker.

(e) Detailed Course Content

- **Module 1: Overview of Cloud computing (6 CH)** Cloud architecture and benefits, enabling technologies, evolution and case studies for motivation, common cloud providers, cloud computing platforms, types and components of clouds, Deployment models i.e public, private and hybrid, Key characteristics, obstacles and opportunities to cloud computing.

- **Module 2: Service Models (6 CH)** Infrastructure as a service (IaaS), Platform as a service (PaaS), Software as a service (SaaS)
- **Module 3: Cloud enabling technologies (12 CH)** Using and managing virtual machines in the cloud (Virtualization). Data centers, broadband, containers. Gain experience with Git, VMWare (or any other hypervisor), Docker.
- **Module 4: Cloud Security (8 CH)** Cloud security challenges, security approaches. Securing the cloud and data. Establishing identity and presence. Data security and control.
- **Module 5: Migrating into the cloud (8 CH)** A case study project with any of the existing services such as Amazon, Google, Microsoft Azure and others.
- **Module 6: Cloud Native Computing and Microservices architectures (5 CH)**. Introduction to cloud native computing, Micro-services, Desining Cloud Native applications, Serverless computing, Containers and Kubernetes, DevOps, continuous integration and delivery.
- **Module 7: Project (15 CH)** Students work in small groups to create an application and be able to host it in the cloud using technologies learned and available cloud platforms such as AWS, GCP, Azure and the local ones such as Crane Cloud.

(d) Mode of Delivery
The teaching pattern is by blended learning including lectures, lab sessions and group projects.

(g) Mode of Assessment
Assessment will constitute practical assignments, tests and a project done in groups. A project assessment will include both a written report and oral presentation (40%) and written Exam (60%).

(h) Reading List

[1] Rajkumar Buyya and James Broberg and Andrzej Goscinski Cloud Computing: Principles and Paradigms, Willey 2011.

[2] Ray J Rafaels Cloud Computing : From beginning to end.

[3] Thomas Erl and Zaigham Mahmood and Ricardo Puttini Cloud Computing: Concepts, Technology & Architecture, Prentice Hall 2013

[4] Boris Scholl, Trent Swanson, Peter Jausovec Cloud Native: Using Containers, Functions, and Data to Build Next-generation Applications, O'Reilly Media, Inc Hall, 2019.

[5] Engineer Bainomugisha and Alex Mwotil Crane Cloud: A Resilient Multi-Cloud Service Layer for Resource Constrained Settings, 2021.

5.5.6 Automata, Complexity and Computability

Course Name:	Automata, Complexity and Computability
Course Code:	CSC 2210
Course Credit:	3
Contact Hours:	45
Year of Study:	2
Semester:	2

(a) Course Description
The course covers an introduction to the classical and contemporary theory of computation including automata, formal languages, Turing machines, recursive functions, computability and in computability, complexity, and the classes of P and NP. It examines the classes of problems that can and cannot be solved by various kinds of machines.

(b) Course Objectives
The objectives of this course are to educate the students in:

- The concepts, notations, and techniques of the theories of Automata, Formal Languages, and Turing machines
- Classification of computational problems given to them, in terms of their computational complexity
- Deeper appreciation for some of the fundamental issues in computing that are independent of trends of technology, such as the Church-Turing Thesis and the P versus NP problem.

(c) Learning Outcomes
By the end of this course, students will be able to:

- know different descriptions of various classes of formal languages.
- prove regularity or context-freeness of languages and non regularity or non context freeness of languages.

- define automata, pushdown automata and Turing machines which perform prescribed tasks.
- know basic unsolvable problems and will be able to prove unsolvability by means of reductions.
- know the basics of the complexity classes P and NP, and of NP1 complete problems.

(e) Detailed Course Content

- **Module 1: Introduction (4 CH)** definitions, why study automata, complexity and computability, applicability and relevance, computation theory, complexity theory, automata theory, languages and strings.
- **Module 2: Finite state machines and regular languages (15 CH)** finite automata, regular expressions, applications of regular expressions, deterministic and nondeterministic finite automata, transition diagrams, equivalence of NFAs and DFAs, conversion from a regular expression to a DFA, conversion of finite automata to regular expression, equivalence and minimization of automata.
- **Module 3: Context Free Grammars (10 CH)** definition of CFG, examples, derivations, sentential forms, parse trees, applications of CFGs, ambiguity in grammars and languages.
- **Module 4: Push-down automata (6 CH)** definition and examples, equivalence of PDAs and context-free grammars i.e A PDA from a Given and A CFG from a Given PDA, deterministic PDAs, Top-down and bottom-up parsing
- **Module 5: Computability (5 CH)** Turing machines, Church-Turing Thesis, Undecidable Problems
- **Module 6: Complexity (5 CH)** The classes P and NP, NP-Computable problems, time complexity - big O and small O notations, decision trees.

(d) Mode of Delivery
Teaching will be in form of class blended learning including lectures, tutorials and group assignments.

(g) Mode of Assessment
Assessment will be in terms of Assignments and tests (40%) and final written examination (60%)

(h) Reading List

[1] Elaine Rich. Automata, Computability and Complexity: Theory and Applications.

[2] Michael Sipser. Introduction to the Theory of Computation, Third Edition, 2013.

[3] John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman. Introduction to Automata Theory, Languages, and Computation. Addison Wesley, 2000.

[4] Daniel I. A. Cohen. Introduction to Computer Theory. John Wiley & Sons, Inc., 1986.

5.5.7 Software Construction

Course Name:	Software Construction
Course Code:	**CSC 2118**
Course Credit:	4
Contact Hours:	60
Year of Study:	2
Semester:	2

(a) Course Description
The course introduces the fundamental processes, principles and techniques of software development. It focuses on development of software that is safe from bugs, easy to understand and maintainable. Topics include tools and techniques for software construction, unit and integration testing, debugging and qualities of software systems such as correctness, clarity, reliability, efficiency and portability. The course provides a hands on experience with a project at the end of the course.

(b) Course Objectives
The objectives of this course are to educate the students in:

- Fundamental principles of software development.
- Software testing and debugging.
- Object-oriented design with UML tools.

(c) Learning Outcomes
By the end of the semester, students will:

- be conversant with program design patterns.
- demonstrate a working knowledge of the object-oriented paradigm for software design and implementation.
- be able to apply methods of systematic testing.
- demonstrate knowledge of agile programming
- have a broader and deeper knowledge of building software systems
- be knowledgeable in version control and use of frameworks for software development

(e) Detailed Course Content

- **Module 1: Preliminaries (6 CH)** Definition and fundamentals, Software Construction activities, why study software construction, standards, planning for software construction. Software development process models including the waterfall model, incremental development and Reuse-oriented software engineering. Code documentation and Code Review - purpose, smelly code, style standards, fail fast.
- **Module 2: Configuration management and Version control (8 CH)** incremental development, repositories and working copies, distributed versus centralized, features and best practices, Git and open source online repositories, team version control, Git in IDEs, Software development tools (IDES), Frameworks and developing graphical user interfaces.
- **Module 3: Debugging and Testing (8 CH)** planning and testing strategies, finding and fixing bugs, techniques for avoiding debugging, strategies for debugging, debugging tools; Testing: unit and integration testing, test case generation; Blackbox and Whitebox Testing. Testing procedures and tools; Error-Handling Techniques; Defensive programming. the software testing process, Test management and automation; Alpha and Beta Testing; testing and quality assurance; functional versus structural testing; Testing and the Software Life Cycle; Static testing: techniques and types; Dynamic testing
- **Module 4: Object-oriented design (10 CH)** Introduction to UML, use cases and scenarios, use case diagrams, UML diagrams: sequence, object and class, deployment diagrams, using UML tools. Designing good APIs; API development.
- **Module 5: Code Design (8 CH)** the design process - inputs, activities and outputs. Minimizing and managing complexity: the software crisis,

separation of concerns. Desirable design characteristics, levels of software design, design techniques, design approaches including iteration, divide and conquer, prototyping, collaborative design.

- **Module 6: Design patterns for Object-oriented programming (10 CH)** when (not) to use patterns; Behavioral Patterns e.g observer, visitor patterns; Structural patterns such as adapter and decorator patterns; Creational patterns such as factory, prototype patterns.
- **Module 7: Agile software development (5 CH)** agile methods, extreme programming; project management.
- **Module 8: Formal Verification (5 CH)** Basic Concepts: Formal Verification and Formal Specification; Axioms; Assertions and correctness proofs: pre- and post-conditions; Weakest pre-conditions; Software inspection and walkthroughs - informal code reviews, formal inspection;

(f) Study Materials
Text books, conference and journal publications, and online resources.

(d) Mode of Delivery
The teaching pattern is by blended learning including lectures, tutorials, practical lab work, group discussion and class presentations.

(g) Mode of Assessment
A programming project will be given that covers all topics learned in the course. The assessment will be done by weekly programming assignments, quizzes and tests (40%) and final examination (60%).

(h) Reading List

[1] Steve McConnell. Code Complete: A Practical Handbook of Software Construction, Second Edition. *Microsoft Press*, 2004.

[2] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software 1st Edition. *Neural Computation*, 74(9):1429–1435, April 2011.

[3] Brian W. Kernighan and Rob Pike The Practice of Programming, 1999.

(a) Course Description
The course introduces the fundamental processes, principles and techniques of software development. It focuses on development of software that is safe from bugs, easy to understand and maintainable. Topics include tools and techniques

for software construction, qualities of software systems such as correctness, clarity, reliability, efficiency and portability.

(b) Course Objectives
The objectives of this course are to educate the students in:

- fundamental principles of software development.
- software testing and debugging.
- Object-oriented design with UML tools.

(c) Learning Outcomes
By the end of the semester, students will:

- be conversant with program design patterns.
- demonstrate a working knowledge of the object-oriented paradigm for software design and implementation.
- be able to apply methods of systematic testing.
- demonstrate knowledge of agile programming
- have a broader and deeper knowledge of building software systems
- be knowledgeable in version control and use of frameworks for software development

(e) Detailed Course Content

- **Module 1: Preliminaries (6 CH)** Definition and fundamentals, Software Construction activities, why study software construction, standards, planning for software construction. Software development process models including the waterfall model, incremental development and Reuse-oriented software engineering. Code documentation and Code Review - purpose, smelly code, style standards, fail fast.
- **Module 2: Configuration management and Version control (8 CH)** incremental development, repositories and working copies, distributed versus centralized, features and best practices, Git and open source online repositories, team version control, Git in IDEs, Software development tools (IDES), Frameworks and developing graphical user interfaces.
- **Module 3: Debugging and Testing (8 CH)** finding and fixing bugs, techniques for avoiding debugging, strategies for debugging, debugging tools; Testing: unit and integration testing, test case generation; Black

box and White box Testing. Testing procedures and tools; Assertions and correctness proofs: pre- and post-conditions; Error-Handling Techniques; Defensive programming.

- **Module 4: Object-oriented design (10 CH)** Introduction to UML, use cases and scenarios, use case diagrams, UML diagrams: sequence, object and class, deployment diagrams, using UML tools. Designing good APIs; API development.
- **Module 5: Code Design (8 CH)** the design process - inputs, activities and outputs. Minimizing and managing complexity: the software crisis, separation of concerns. Desirable design characteristics, levels of software design, design techniques, design approaches including iteration, divide and conquer, prototyping, collaborative design.
- **Module 6: Design patterns for Object-oriented programming (10 CH)** when (not) to use patterns; Behavioral Patterns e.g observer, visitor patterns; Structural patterns such as adapter and decorator patterns; Creational patterns such as factory, prototype patterns.
- **Module 7: Agile software development (10 CH)** agile methods, extreme programming; project management.

(d) Mode of Delivery
The teaching pattern is by lectures, tutorials, practical lab work, group discussion and class presentations.

(g) Mode of Assessment
The assessment will be done by mini-projects with presentation made in class, tests and assignments (40%) and final examination (60%)

(h) Reading List

[1] Steve McConnell. Code Complete: A Practical Handbook of Software Construction, Second Edition. *Microsoft Press*, 2004.

[2] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software 1st Edition. *Neural Computation*, 74(9):1429–1435, April 2011.

[3] Brian W. Kernighan and Rob Pike The Practice of Programming, 1999.

5.6 Year II Recess Term

5.6.1 Field Attachment

Course Name:	Field Attachment
Course Code:	CSC 2303
Course Credit:	5
Contact Hours:	75
Year of Study:	2
Semester:	Recess Term

(a) Course Description
This course provides students with a mentorship opportunity in a real work environment. Students are given an opportunity to go out in the field to put whatever they have learnt into practice. It is a 2-3 months program and during this period students are attached to various organizations where they carry out training from.

(b) Course Objectives
The aims of the course are;

- To equip students with practical skills in all aspects of computer science such as programming, databases, software engineering, artificial Intelligence and general hardware maintenance.
- To provide students with mentor ship in writing professional software products.
- To give students an opportunity to build team-work, communication, appearance and leadership skills needed in the workforce.

(c) Learning Outcomes
Upon successful completion students will;

- Have gained work experience that allows them to sample professional environments in which they might seek careers.
- Have experience that will help prepare them for careers and research ahead.

(e) Detailed Course Content
Not Specified

(f) Study Materials
Not Specified

(d) Mode of Delivery
The student is to be under the supervision of one of the workers (Field Supervisor) in the organization. The student is assigned duties in line with the operations of the organization. Staff from Makerere University (Academic Staff) will make visits to get the students view of the organization as well as the organizations view about the student.

(g) Mode of Assessment

- The two supervisors (both Academic and Field Supervisor) will be given forms to evaluate the students. On the side of the student, he/she is expected to send a weekly report to the academic supervisor on the various activities carried out and at the end of the internship provide a final report.
- The two reports will be used to evaluate the student. The final report carries 40 %, the Academic supervisor assessment carries 40% and Field supervisor assessment carries 20%.

(h) Reading List
There is no particular reference material for internship, except whatever may be recommended by the student supervisors.

5.7 Year III Semester I

5.7.1 Entrepreneurship Principles

Course Name:	Entrepreneurship Principles
Course Code:	BAM 2102
Course Credit:	3
Contact Hours:	45
Year of Study:	3
Semester:	1

(a) Course Description
This course teaches how to recognize a business opportunity and develop the opportunity into a business. The controlling functionsaccounting, finance, marketing, managementas well as legal and economic considerations are applied. In this course Student responsibility and initiative are encouraged as business strategies are created, planned, and presented as a final producta business plan

for an actual business venture. Through the process of developing the business plan, students acquire skills necessary to operate a successful business.

(b) Course Objectives
The aims of this course are to help students to:

1. Understand entrepreneurial behavior with reference to business
2. Acquire skills involved in creating, planning, developing and managing the entrepreneurial business.
3. Recognize the importance of entrepreneurship and identify the profile of entrepreneurs and their role in economic growth
4. Identify an attractive market that can be reached economically

(c) Learning Outcomes
After successfully completing this course, students should be able to:

1. Describe entrepreneurial behavior with reference to business
2. Demonstrate knowledge involved in creating, planning, developing and managing the entrepreneurial business
3. Articulate the importance of entrepreneurship and identify the profile of entrepreneurs and their role in economic growth
4. Exhibit the ability to find an attractive market that can be reached economically

(e) Detailed Course Content

- **Module 1: Introduction to Business** (6 Hours)
- **Module 2: Introduction to Entrepreneurship** (6 Hours)
- **Module 3: Business and the entrepreneurial environment** (3 Hours)
- **Module 4: Creating the entrepreneurial business** (6 Hours)
- **Module 5: Business planning** (6 Hours)
- **Module 6: Successful business strategies** (6 Hours)
- **Module 7: Forming and protecting a business Managing the Entrepreneurial business** (3 Hours)
- **Module 8: Management of Resources in the entrepreneurial business** (3 Hours)

- **Module x: 9. Marketing in an entrepreneurial business** (3 Hours)
- **Module 10: Managing Money in an entrepreneurial business** (3 Hours)

(f) Study Materials
Text books, conference and journal publications, and online resources.

(d) Mode of Delivery
The teaching will be conducted using blended learning approaches including:

1. Lectures
2. Take-home assignments
3. E-learning
4. Discussions and presentations
5. Problem-based/case studies

(g) Mode of Assessment
Assessment will be by continuous assessment through tests, assignments (40%) and final exam (60%).

(h) Reading List

[1] Stokes, D.R. & Wilson, N.. Small Business Management and Entrepreneurship, 6th Edition. Publisher: Carnegie Learning EMEA, 2010.

[2] Storey, D.J. & Greene, F.J. Small Business and Entrepreneurship. Illustrated Edition. Publisher: Financial Times Prentice Hall., 2010

[3] Beaver, G. Small Business, Entrepreneurship & Enterprise Development.. Publisher: Financial Times Prent. Int., 2002.

5.7.2 Business Process Management

Course Name:	Business Process Management
Course Code:	IST 3110
Course Credit:	4
Contact Hours:	60
Year of Study:	3
Semester:	1

(a) Course Description

This course introduces the fundamentals of business process management (BPM) that will be used to systematically analyze, improve and automate business processes. Students will learn the techniques and tools that will be used for process identification, process discovery, process analysis, process redesign and improvement, process automation and monitoring. Students will learn how to apply these techniques and tools to a wide range of examples and case studies that will show the power of BPM in practice.

(b) Course Objectives

The aims of the course unit are to:

- Equip students with skills needed to identify business processes within an organization.
- Equip students with skills necessary to document business processes using different business process modeling techniques.
- Make students understand qualitative and quantitative techniques useful for the analysis of the performance of business processes.
- Equip students with skills necessary to identify and analyze business improvement opportunities based on business process models.
- Make students understand how IT can effectively be used to build process automation solutions that realise the benefits of process improvement.

(c) Learning Outcomes

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

- Identify business processes within an organization, study their boundaries and interrelations, and prioritize their management using different criteria.
- Document business processes at different levels of detail using contemporary business process modeling techniques.
- Apply qualitative and quantitative techniques to analyze the performance of business processes and to assess the impact of business process changes
- Identify and analyze business improvement opportunities based on business process models.
- Use IT effectively to build process automation solutions that realise the benefits of process improvement.

(e) Detailed Course Content

- **Module 1: Introduction to Business Process Management (5 CH)**
  Introduction to the definition of business process and BPM. Provide a historical overview of the BPM discipline. Finally, the BPM lifecycle will be discussed.
- **Module 2: Process Identification (5 CH)**
  Discuss the context of process identification. Present a method based on process architecture definition and process selection with listing initial set of processes and their architecture. Discuss process selection which considers criteria for defining priorities of processes using a portfolio.
- **Module 3: Essential Business Process Modelling (10 CH)**
  Discuss business process modelling based on the Business Process Modelling Notation (BPMN). Discuss branching and merging, business objects, and resources. Cover process decomposition and process model reuse.
- **Module 4: Process Discovery (10 CH)**
  Discuss the setting of process discovery and process discovery methods. Process Modelling method and Process Model Quality Assurance.
- **Module 5: Qualitative Process Analysis (5 CH)**
  Discuss value added analysis and waste analysis. Stakeholder Analysis & Issue Documentation. Root-Cause Analysis.
- **Module 6: Quantitative Process Analysis (15 CH)**
  Cover topics on Flow Analysis, Queuing Analysis and Process Simulation.
- **Module 7: Process Redesign (5 CH)**
  Discuss the essence of process redesign. Transactional Methods to process redesign and Transformational Methods.
- **Module 8: Process-Aware Information Systems (5 CH)**
  Types of Process-Aware Information Systems, Advantages of Introducing a BPMS, Challenges of Introducing a BPMS

(d) Mode of Delivery
Blended learning including:

- Lectures.
- Practical sessions in the computer laboratories.
- Online learning management systems.
- Class discussions and presentations.

- Problem-based/case studies.
- Project-based assignments that can be done in groups.

(g) Mode of Assessment
The assessment will be in form of:

- Tests 20%
- Individual assignment 20%
- Final written exam 60%

(h) Reading List

[1] Dumas, M. La Rosa, M., Mendling J., & Reijers, H. A. Fundamentals of Business Process Management (2nd Ed) Springer., 2018.

[2] Weske, M. Business Process Management: Concepts, Languages, Architecture. (3rd Ed) Springer, 2019

5.7.3 Cryptology and Coding Theory

Course Name:	Cryptology and Coding Theory
Course Code:	CSC 3114
Course Credit:	3
Contact Hours:	45
Year of Study:	3
Semester:	1

(a) Course Description
This course provides a foundation for further studies in information security. It introduces students to the exciting fields of cryptology and coding theory. Fundamentally, it deals with the mathematics that underlies modern cryptology. Cryptology combines the studies of cryptography, the creating of masked messages, and cryptanalysis, the unraveling of masked messages. Coding theory is the study of coding schemes used to detect and correct errors that occur during the data transmission. Basic problems of cryptography and coding are discussed. Topics include classical ciphers, public key cryptosystems (RSA, Diffie-Hellman key exchange, ElGamal), digital signatures, codes, linear codes, perfect codes and cyclic codes.

(b) Course Objectives

The objectives of this course are to educate the students:

- To understand the building blocks of crypto systems and error correction
- To gain historical understanding of the evolution of crypto systems.
- To develop tools necessary to crypto analyze crypto systems
- To gain insights in the practical application of cryptology and error correction in the modern information age.
- To understand the goals and trade-offs associated with encryption and error-control coding systems.

(c) Learning Outcomes

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

- Deploy sound cryptographic practices and tools
- Discuss the goals and trade-offs associated with encryption and error-control coding systems.

(e) Detailed Course Content

- **Module 1: Overview (5 CH)**
  History of cryptology and coding theory; Classical Cryptography and cryptosystems
- **Module 2: Encryption (10 CH)**
  Symmetric and asymmetric; Private-Key and Public-Key Cryptography; Key management - Diffie-Hellman Key Exchange, Elliptic curve cryptography; Digital Signatures - RSA algorithm; Steganography
- **Module 3: Block Ciphers (8 CH)**
  Data Encryption Standard (DES); Advanced Encryption Standard (AES); Secure Communication and Message Integrity; Message Authentication Codes and hash functions. Stream Ciphers.
- **Module 4: Security Protocols (10 CH)**
  Authentication protocols - mutual authentication and one-way authentication; IP security; Secure Sockets Layer (SSL), SSL/TLS; Email security; Security Facilities in the TCP/IP Protocol Stack.
- **Module 5: Coding theory (12 CH)**
  Coding theory fundamentals; Error detection, correction and decoding, Error-correcting codes: Hamming distance, Linear codes; Hamming codes, Reed-Muller codes; Cyclic codes; Applications in computer science.

(d) Mode of Delivery
The course will be delivered by blended learning including lectures, tutorials, and group assignments.

(g) Mode of Assessment
Assessment will be by assignments and tests (40%) and written examination (60%)

(h) Reading List

[1] A. Menezes, P. van Oorschot, and S. Vanstone. Handbook of Applied Cryptography. CRC Press, 1996.

[2] Trappe and Washington. Introduction to Cryptography with Coding Theory, 2nd edition. Prentice Hall, 2006.

[3] Simon Singh. The Code Book. Doubleday, 1999.

[4] William Stallings. Cryptography and network security: principles and practice (7th edition). Pearson, 2017.

5.7.4 Advanced Programming

Course Name:	Advanced Programming
Course Code:	CSC 3115
Course Credit:	4
Contact Hours:	60
Year of Study:	3
Semester:	1

(a) Course Description

This course highlights programming practices that are vital in the day today work of a programming Professional. While many systems are described by functionalities, some important aspects like security, robustness, and maintainability are ignored. Students are to get an in depth understanding of these concepts as well as exploring the current trends in the programming environment. In the first half of the course, a framework like Django, CodeIgniter, Ruby on Rails, Laravel (or Symfony), MonoRail, or Bootstrap is to be used to demonstrate the MVC (Model-View-Controller) architecture. In the other half, the course should also provide mobile applications development skills to the learners.

(b) Course Objectives

The aim of the course is to concretize the student's past programming experience as well as highlighting critical practices like efficient code reuse and parallel development in programming that are necessary for a professional programmer

(c) Learning Outcomes

By the end of the course, the student should be:

- Able to develop well documented and well- structured software that can easily be maintained
- Able to implement a complete application using the MVC architecture
- Knowledgeable in other programming practices like mobile applications development
- Aware of newer programming paradigms like service oriented and cloud computing

(e) Detailed Course Content

- Module 1 (8 hours): **The MVC architecture**
- Module 2 (8 hours): **Regression test programming**
- Module 3 (12 hours): **Trends in programming paradigms**
- Module 4 (16 hours): **Visualisation**
- Module 5 (16 hours): **Mobile applications development**

(d) Mode of Delivery

There will be blended (online or face-to-face) sessions, tutorials, and/or practical sessions.

(g) Mode of Assessment

The assessment will be done by tests and take home practical assignments (40%), practical examination (30%) and written examination (30%)

(h) Reading List

1. Advanced Programming in the UNIX Environment by W. Richard Stevens and Stephen A. Rago Addison Wesley 2013
2. Test-Driven Development: A Practical Guide, by David Astels
3. Beautiful Code: Leading Programmers Explain How They Think, by Andy Oram & Greg Wilson
4. Refactoring: Improving the Design of Existing Code, by Martin Fowler
5. Refactoring to Patterns, Joshua Kerievsky

5.7.5 Operations Research

Course Name:	Operations Research
Course Code:	CSC 3117
Course Credit:	3
Contact Hours:	45
Year of Study:	3
Semester:	1

(a) Course Description
This course introduces to the student the concepts of operations research (OR), enable the student explore the different manifestations of operations research and go deeper into two such manifestations: Linear systems and graphical systems. The students will learn how to use popular OR packages such as GAMS, AMPL, and OPL.

(b) Course Objectives
The objectives of the course are to enable students understand

1. What Operations Research is and its use in general life
2. How to formulate and solve simple linear programming problems
3. How to analyse effects of variations in LP problems including modifications of the initial formulation
4. How to solve simple graphical problems
5. How to apply graphical approaches (like matching, traversal and spanning trees) in real life /Computing scenarios and use of OR programming packages.

(c) Learning Outcomes

By the end of the course, the learner shall be able to

1. Explain what OR is, its variants and its role in real life
2. Formulate LP problems given the rubric of the problem
3. Solve LP problems and be able to deduce special cases (like in existent solutions, unbounded solutions, degeneracy and alternate optima)
4. Be able solve basic graphical problems
5. Be able to interpret graphical problems in relation to real life and use popular OR computing packages/libraries.

(e) Detailed Course Content

- **Concepts of Operations Research:** (10 CH)
  Introduction to concepts of operations research, a classical example of a real life problem that can be solved in an Operations Research way at minimal resources (time, space, money, materials, etc). Discussion of 2-3 typical generic operations research problems, how they apply in real world and instances of solutions to them. The generic problems can be any of the likes of (i) Knapsack problem (ii) traveling salesman problem (iii) Chinese postman problem (iv) Celebrity problem (v) Stable marriage problem etc
- **Introduction to Linear Programming:** (10 CH)
  Formulation of LP problems, graphical solutions to LP problems, simplex approach, artificial variables, big M and duality
- **Further Linear Programming:** (10 CH)
  Algebraic LP solutions, Properties of LP problems/solutions, Post optimality Analysis, sensitivity analysis (and their implication in real life)
- **Graph Related Systems:** (15 CH)
  Graph definition, representation, properties, types, graph traversal, (minimum) spanning trees, matching and matching problems, graph coloring. The applications to the world of computer science will be included. Solving OR problems in computing packages such as GAMS, AMPL, OPL, available for programming environments such as Python.

(f) Study Materials

Materials will include text books, lecturers slides, online resources and students discussions

(d) Mode of Delivery
The mode of study will be by blended learning including a mixture of student and teacher centered. The teacher will explain the concepts in class but students will also make extensive discussions and guided reading

(g) Mode of Assessment
The assessment shall include at least one test and an assignment per module - that shall make progressive assessment (40%) as well as a final examination that shall contribute 60%

(h) Reading List

[1] Hamdy A. Taha Operations Research : An Introduction, 2015.

[2] Frederick Hillier Introduction to Operations Research . 10th Edition

5.7.6 Computer Science Project 1

Course Name:	Computer Science Project 1
Course Code:	CSC 3118
Course Credit:	5
Contact Hours:	75
Year of Study:	3
Semester:	1

(a) Course Description
The Computer Science Project 1 is a 5 credit unit course that aims at providing a platform for students to combine knowledge and skills acquired from all course modules taught throughout the course in the production of a suitable Computer science project. The project is not a taught module but a major design and development exercise for the student carried out under supervision.

The students in this course will select a project topic from an area of computer Science that interest them, conduct a semester worth of individual/group study of that topic, resulting into a substantial written proposal and software model or solution demonstrating initial preliminary results from the study. The project should be of a problem solving nature, typically providing a software solution or a model to a practical computing problem.

The project should also be able to produce an end product for users. Further, theoretical essay, literature search, or a descriptive evaluation would not be acceptable.

(b) Course Objectives

- To give experience of undertaking a significant individual/group design and development exercise from conception through to design, implementation and delivery.
- To allow students to conduct a substantial piece of research into unfamiliar area of computing.
- To introduce research methods and skills relevant for conducting such a project.
- To provide an opportunity for the students to synthesize what they have learned from other modules.
- To teach students planning, scientific writing and problem solving skills.
- To give students experience of working independently.
- To give students experience in writing up and presenting work in a scholarly fashion.

(c) Learning Outcomes
On successful completion of this course students will be able to.

- A general understanding of the state of the art in the student's chosen research area of specialization.
- Have an in-depth knowledge and understanding of the student's chosen area of specialization.
- have an understanding of the research methods required in conducting a research based project.
- Plan and carry out research appropriated to a project.
- Demonstrate project planning, scientific writing and problem solving skills.
- Manage personal learning and exercise critical self-evaluation.
- Have effective communication of general and specialized Computing Science concepts

(e) Detailed Course Content

Since the project is not a taught module but a major design and development exercise for the student carried out under supervision; students are expected to select a research project from the topic of interest. The following are some of the topics students can choose from

- Data Warehousing
- Internet of Things (IoT)
- Big Data/Data science
- Semantic Web
- Machine Learning
- Artificial Intelligence
- Data Mining
- Image Processing
- Bioinformatics
- Operating systems
- Multimedia computing
- Computer vision
- Algorithms & complexity

(d) Mode of Delivery

The course is a self-study/group-study research exercise for the students carried out under supervision.

(g) Mode of Assessment

A detailed project proposal is submitted to the supervisor in the 14th week of the first semester for 40% (combined with student participation). A poster presentation of preliminary results of the project and a well written project proposal are presented to a panel of examiners for 60%.

(h) Reading List

[1] Hossein Hassani. How to do the Final Year Projects. A Practical Guideline for Computer Science and IT Students.

[2] Ranjit Kumar. Research Methodology a step-by-step guide for beginners 3rd *Edition, 2014.*

[3] C.R. Kothari. Research Methodology methods and & techniques *Second Edition, 2004.*

5.7.7 User Interface Design

Course Name:	User Interface Design
Course Code:	CSC 3119
Course Credit:	4
Contact Hours:	60
Year of Study:	3
Semester:	1

(a) Course Description
The course introduces the principles of user interface development, focusing on design, implementation and evaluation.

(b) Course Objectives
The course aims at:

- Training students in order for them to create efficient, flexible, usable and interactive User Interfaces (UI)
- Providing students with the skills so that they are able identify system users, the tasks they want to carry out and the environment in which they will be working.
- Exposing students to standard conceptual designing.
- Enable students create useful User interfaces, in particular graphical user interfaces.
- Instructing students in the art of evaluating UIs;

(c) Learning Outcomes
By the end of this course, students will be able to:

- Develop efficient, flexible, usable and interactive User Interfaces(UI)
- Identify system users, the tasks they want to carry out and the environment in which they will be working;
- Create standard conceptual designs.
- Create useful User interfaces, in particular graphical user interfaces.
- Evaluate UIs

(e) Detailed Course Content

- Usability, Design Principles and User-Centered Design **6 Hours**
- UI Software Architecture, Human Capabilities **6 Hours**
- Output Models, Input Models, Conceptual Models and Metaphors **6 Hours**
- Designing for collaboration and communication **6 Hours**
- Emotional interaction **6 Hours**
- The process of interaction design **6 Hours**
- Design, prototyping and construction **6 Hours**

- Graphic Design **6 Hours**
- UI Evaluation **6 Hours**
- User Testing, Experiment Design and Analysis **6 Hours**

(f) Study Materials
Text books, conference and journal publications, and online resources.

(d) Mode of Delivery
The study mode will by blended learning including Lectures, Lab sessions, Online learning management systems, Class discussions and presentations, and Group project-based assignments

(g) Mode of Assessment
Assessment will be in form of Coursework (Assignments and tests) 40% and Final written exam 60%

(h) Reading List

[1] Rogers, Y., Sharp, H., & Preece, J. Interaction design: beyond human computer interaction, 2011

[2] Norman, D. A. The design of everyday things: Revised and expanded edition. Basic books, 2013.

5.7.8 Computer Graphics

Course Name:	Computer Graphics
Course Code:	CSC 3121
Course Credit:	3
Contact Hours:	45
Year of Study:	4
Semester:	1

(a) Course Description
The course provides an introduction to the theory and practice of computer graphics; specifically it introduces the basics of 2- and 3-dimensional computer graphics where both the theoretical principles and methods will be taught and how they can be applied using graphical software libraries. More to that students will learn about the theoretical underpinnings of computer graphics and

practical know-how needed to quickly and effectively apply these techniques to new problems.

Topics will include: Introduction to Computer Graphics and OpenGL, Basic Programming Techniques using OpenGL, graphics systems, graphics devices and their control, 2D graphics, Color Systems and Shading,geometric object, Transformation and Viewing, Objects Modeling and Visible Surface Detection, clipping and hidden surface removal, Lighting, Surface Rendering, Basic Ray Tracing Algorithms, Applying Ray Tracing Techniques,

(b) Course Objectives

- To Identify and explain the core concepts of computer graphics.
- To Apply graphics programming techniques to design, and create computer graphics scenes.
- To create effective OpenGL programs to solve graphics programming issues, including 3D transformation, objects modeling, colour modeling, lighting, textures, and ray tracing.

(c) Learning Outcomes
On successful completion of this course students will be able to.

- describe the data flow in a graphics rendering system.
- derive and apply geometric view and projection models and transformations of homogeneous coordinates in computer graphics, like transformations of 3D objects, transformations between object-world-camera coordinate systems, and perspective and parallel projections.
- describe how lines, surfaces and in some cases curves can be represented by polygons and parametric curves, as well as be able to derive the definition and use these representations.
- derive and apply basic rendering techniques and algorithms in polygon-oriented computer graphics such as lightning models, line and polygon cutting algorithms, and how to treat hidden surfaces.
- describe and relate various visual effects such as antialiasing, texture mapping, bump mapping, and displacement mapping.
- demonstrate their ability to use modern 3D computer graphics techniques, models, and algorithms to solve graphics problems.

(e) Detailed Course Content

- **Module 1: Introduction to computer graphics, problem domain and application. (3 hours)**
- **Module 2: Coordinate systems and Transformations. (9 hours)**
  The module discusses how various graphics can be changed into some thing else by applying a set of rules or transforms. The module will discuss 2D and 3D transforms, geometric operations and hierarchical 3D transformations.
- **Module 3: Building and representing. (09 hours)**
  This module will discuss various ways of representing 3D objects or surfaces in computer graphics. We will discuss data sources and acquisition, modeling software and representation schemes.
- **Module 4: The Computer graphics pipeline and the OpenGl API for 3D computer graphics. (3 hours)**
- **Module 5: Rendering . (9 hours)**
  We discuss various ways of rendering objects. These will include incremental shading algorithm, Hidden Surface removal algorithms, the Z-buffer and the Ant-Anti-aliasing algorithms .
- **Module 6: Simulating 3D Reality. (9 hrs)**
  This module discuss different ways of simulating real-world objects in computer graphics. We will discuss light object interaction, texture and shadows.
- **Module 7: Introduction to Computer Animation. (3 hours)**

(f) Study Materials
Textbooks, Assignments,projects, online materials, e-learning

(d) Mode of Delivery

- Teaching will be conducted using blended learning approaches including lectures, tutorials, quizzes,
- Assignments and project work: The best way to learn about a computer graphics method is to program it yourself and experiment with it. So the assignments will generally involve implementing computer graphics algorithms, and experimentation to test your algorithms on some data. The students may asked to summarize their work, and analyze the results, in brief (3-4 page) write ups. The implementations will be done in OpenGL.

(g) Mode of Assessment
Assessment will be by continuous assessment through programming assignments (20%) and written assignments and quizzes (20%) and final exam (60%).

(h) Reading List

[1] Peter Shirley and Steve Marschner. Fundamentals of Computer Graphics. Third Edition, 2009.

[2] Edward Angel and Dave Shreiner . Interactive Computer Graphics A top-down Approach with shader-based OpenGL. 6^{th} edition, 2012.

[3] J. F. Hughes and A. van Dam and M. McGuire and D. F. Sklar and J. D. Foley and S. K. Feiner and K. Akeley Computer Graphics Principles and Practice. 3^{rd} Edition 2014.

5.7.9 Modelling and Simulation

Course Name:	Modelling and Simulation
Course Code:	IST 3208
Course Credit:	3
Contact Hours:	45
Year of Study:	3
Semester:	1

(a) Course Description
This course provides an introduction to system modeling using both computer simulation and mathematical techniques. A range of case studies are examined in the lectures and project exercises. Students will get hands-on training. The application areas considered are wide-ranging, although the emphasis is on the analysis of a variety of modeling paradigms such as queuing and dynamic systems. A simulation language will be utilized and an applied project carried out.

(b) Course Objectives
The aims of this course are to help students to:

- Acquire basic knowledge about simulation and modeling
- Develop critical thinking and analytical skills
- Attain skills of building simulation models and learn how to model and simulate a variety of management-related problems.
- To be able to solve real world problems which cannot be solved strictly by mathematical approaches using Modeling and Simulation.

(c) Learning Outcomes
By the end of the course students should be able to:

- Demonstrate basic knowledge about simulation and modeling
- Exhibit critical thinking and analytical skills
- Display skills of building simulation models and how to model and simulate a variety of management-related problems.
- Manifest competence in solving real world problems which cannot be solved strictly by mathematical approaches using Modeling and Simulation.

(e) Detailed Course Content

- **Introduction to simulation concepts** : simulation, models, benefits of using simulation, application areas, model classifications, types of simulations, benefits and drawbacks of simulation, Pitfalls to successful completion of simulations.(3 Hours)
- **Introduction to systems-** static and dynamic; discrete and continuous, Introduction to Monte Carlo simulation, generation of randomm numbers using Excel RAND () as well as various distributions such as, Bernoulli, Exponential. (6 hours)
- **Discrete Event Simulation (DES)**: components of DES, overview of Queuing Theory, queuing systems, queuing models, queuing discipline (LIFO, FIFO, SIRO, priority queue), analysis and performance measures of queuing systems, overview of probability distributions (such as Markovian, deterministic) used in queuing models. (6 Hours)
- **Hand simulation exercises** employing single/multi-channel, single/multiphase server systems (6 Hours)

- **Simulation Development Life Cycle**: problem formulation, system investigation, model formulation, model representation, programming, design of experiments, experimentation, presentation of simulation results, verification and validation. (6 Hours)
- **Graphical Integration:** basic principles and application of graphical integration, constant rates, inflows and outflows, step functions. (6 Hours)
- **Introduction to dynamic simulation models**: Stock and Flow diagrams, STELLA modeling environment. Exercises involving modeling with STELLA, interpretation of graphs, creating user interfaces and them to the models. (6 Hours)
- **Undertake DES and dynamic simulation field projects**. (6 Hours)

(f) Study Materials
Text books, conference and journal publications, and online resources

(d) Mode of Delivery
Blended learning including:

- Lectures
- Lab practicals
- Online learning management systems
- Class discussions and presentations
- Group project-based assignments

(g) Mode of Assessment
Course work (tests, assignments) 20 % , Projects 20 % and Final written exam: 60%

(h) Reading List

[1] Principles of Modeling and Simulation: A Multidisciplinary Approach Sokolowski, J.A. & Banks, C.M. (2011)). John Wiley & Sons. ISBN: 978-0-470-28943-3.

[2] Modeling and Simulation Fundamentals: Theoretical Underpinnings and Practical Domains. Sokolowski, J.A. & Banks, C.M. (2010). Wiley & Sons. ISBN: 978-0-470-48674-0.

[3] Theory of Modeling and Simulation, (2nd ed) Zeigler, B.P., Praehofer, H., & Kim, T.G. (2000)

[4] Simulation Modeling and Analysis Law, A.M & Kelton, W.D. (2000). , (3rd ed)

[5] Simulation With Arena Kelton, W.D., Sadowski, R.P., & Sturrock, D.T. (2006) (4th ed) Mc-Graw Hill

5.8 Year III Semester II

5.8.1 Data Communications

Course Name:	Data communications
Course Code:	BSE 2206
Course Credit:	4
Contact Hours:	60
Year of Study:	3
Semester:	2
Pre-requisites:	Computer Networks

(a) Course Description
This is a theoretical course that covers the fundamentals of data communication, Formatting and transmission of digital information over various media. The topics covered include digital versus Analog transmission, transmission media, data encoding, integrated Services Digital Networks (ISDN), Network Access Protocols, LAN standards among others.

(b) Course Objectives
The aims of the course are;

- To provide a solid basis on the theoretical and practical understanding of data communication networks.
- To get practical experience of the physical phenomenon that can be used to transmit digital information
- To familiarize the students with the basic taxonomy and terminology of the data communications area.
- To impact knowledge and skill relevant for the design, implementation and maintenance of modern computer communication networks
- To introduce students to emerging technologies in data communication.

(c) Learning Outcomes
Upon successful completion of this course, the student should be able to;

- Learn the design principles of communication networks.
- Describe the theoretical concepts involved in data communication.
- Understand the physical phenomenon that can be used to transmit digital information.
- Describe emerging communication technologies particularly wireless and fiber optic technologies.
- Describe emerging research directions in computer communication networks.
- Work professionally in the area of computer networks and related data communication fields.

(e) Detailed Course Content

- **Module 1: Introduction to Data Communications: (4 CH)**
  What is communication, uses of communication; General block diagram of communication system, types of communication, Data communications, Applications of data communications Data Communications and Networking for Today's Enterprise, A Communications Model, Networks, Internet.
- **Module 2: Data Transmission:(8 CH)**
  Data Transmission: Fourier analysis, Band limited signals, The communication channel, Maximum data rate of a channel, Electromagnetic spectrum, electromagnetic waves, frequency and wave length, bandwidth, bandwidth and channel capacity, Modulation, types of Modulation, Concepts and Terminology, Analog and Digital Data Transmission, Transmission Impairments, Channel Capacity
- **Module 3: Transmission Media:(8 CH)**
  Transmission Media: Guided Transmission Media, Wireless Transmission, Wireless Propagation, Line-of-Sight Transmission, Optical Fibre -Physics & velocity of propagation of light, Advantages & disadvantages, unguided media: Electromagnetic polarization, attenuation and absorption, optical properties of radio waves, terrestrial propagation of electromagnetic waves, skip distance free-space path loss, microwave, infrared & satellite communication system.

- **Module 4: Signal Encoding and Data Communication Techniques: (20 CH)**
  Digital Data, Digital Signals; Digital Data, Analog Signals; Analog Data, Digital Signals; Analog Data, Analog Signals.
  Digital Data Communication Techniques: Digital communication, advantages of digital communication, Nyquist theorem ,Sampling Theory, Analog to digital conversion -Pulse Code Modulation (PCM), Delta modulation (DM); encoding of digital signals, Multiplexing and Modulation of Digital Signals, digital radio, digital amplitude modulation, frequency shift keying (FSK), phase shift keying (PSK), quadrature amplitude modulation (QAM), band width efficiency, carrier recovery, differential phase shift keying,(DPSK), clock recovery, probability of error & bit error rate, trellis encoding, Asynchronous and Synchronous Transmission.
- **Module 5: Spread Spectrum: (8 CH)**
  The Concept of Spread Spectrum, Frequency Hopping Spread Spectrum, Direct Sequence Spread Spectrum, Multiple Access- Random Access, Aloha-Carrier Sense Multiple Access (CSMA)- Carrier Sense Multiple Access with Collision Detection (CSMA)- Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA), - Frequency- Division Multiple Access (FDMA), Time - Division Multiple Access (TDMA), - Code - Division Multiple Access (CDMA).
- **Module 6: Local Area Network Overview and ISDN:(12 CH)**
  Background, Topologies and Transmission Media, LAN Protocol Architecture, Bridges, Hubs and Switches, Virtual LANs, integrated services digital network (ISDN), Network Access Protocols; Passive versus dynamic allocation.
  Ethernet: Traditional Ethernet, High-Speed Ethernet, IEEE 802.1Q VLAN Standard. LAN standards:802.3 (Ethernet),802.4 (token bus), 802.5 (token ring);
  Wireless LANs: Overview, Wireless LAN Technology, IEEE 802.11 Architecture and Services, IEEE 802.11 Medium Access Control, IEEE 802.11Physical Layer, IEEE 802.11 Security Considerations.

(f) Study Materials
Textbooks, Research papers among others

(d) Mode of Delivery
This course will be delivered through online or face to face class lectures and discussions. Simulations will also be used to explain abstract concepts.

(g) Mode of Assessment
(Assignments, Tests, Group Research course work) 40% and Final Examination 60%

(h) Reading List

[1] Behrouz A Forouzan, Data Communications and Networking, 5th Edition (2012)

[2] Kurose, JF & Ross, Computer Networking: A Top Down Approach, 4th edn, Addison-Wesley, 2007

[3] William Stallings, Data and Computer Communications, 7th Edition (2003).

[4] David Stamper et al, Business Data Communications, 6th Edition, 2003, Prentice Hall.

5.8.2 Computer Security

Course Name:	Computer Security
Course Code:	CSC 3207
Course Credit:	3
Contact Hours:	45
Year of Study:	3
Semester:	2

(a) Course Description
Computer security is a branch of technology concerned with digital security or information security applied to computers. Since the largest part of the computer that users interact with is software, computer security pays big attention to development of secure software. The course gives tools for achievingparticularsecuritygoals, attacks, and countermeasures, an overview of security issues for software, and provides programming methods for the development of secure applications.

(b) Course Objectives
The objectives of this course are to educate the students in:

- the need for computer security.
- threats faced by computers in the connected digital world.

- techniques that are used to protect computers against various threats.

(c) Learning Outcomes
By the end of the course, students should be able to:

- Describe digital security principles
- Describe Access control mechanisms
- Apply computing forensic techniques using open-source tools

(e) Detailed Course Content

- **Module 1: Overview of Computer Security (4 CH)**
  Digital security principles - Confidentiality, Integrity, Availability; threats; goals of security; social engineering, Usability; Security engineering; Security architecture.
- **Module 2: Security policies (8 CH)**
  Confidentiality policies - Bell-LaPadula Model; Integrity Policies - Biba Integrity Model, Clark-Wilson Integrity Model; Hybrid Policies - Chinese Wall Model; Multilevel security;
- **Module 3: Authentication and Access Control (6 CH)**
  Basics, Identity representation; Passwords; Challenge-Response; biometrics; location; access control mechanisms and types; access control lists; access control matrix; capabilities.
- **Module 4: Software Security (8 CH)**
  Security by design - Design Principles; Secure coding; Malicious logic; Proof carrying code; confinement problem - virtual machines and sandboxes; Building Secure and trusted systems; Buffer overflow exploits; Securing web applications.
- **Module 6: Risk Management (6 CH)**
  Risk and threat modeling; vulnerability analysis. Compliance: Audits reviews and inspection, Vulnerability scanners, Penetration testing
- **Module 7: Network and Cloud Security (5 CH)**
  Intrusion detection; mobile phones; database security; web security; attacks and prevention.
- **Module 8: Computer Forensics (8 CH)**
  File system analysis; malware analysis; mobile and Internet forensics. Have a practical session on this.

(d) Mode of Delivery
The teaching pattern is by blended learning including lectures, lab sessions and group projects.

(g) Mode of Assessment
Assessment will constitute practical assignments, written course work (40%) and written Exam (60%).

(h) Reading List

[1] Ross J. Anderson. Security Engineering: A Guide to Building Dependable Distributed Systems, Willey 2001.

[2] Dan Farmer and Wietse Venema Forensic Discovery, Addison-Wesley, 2005.

[3] K. M. Jackson and J. Hruska and Donn B. Parker. Computer security reference book, 1992

[4] Matt Bishop Introduction to Computer Security. Addison-Wesley, 2005.

5.8.3 Compiler Design

Course Name:	Compiler Design
Course Code:	CSC 3205
Course Credit:	3
Contact Hours:	45
Year of Study:	3
Semester:	2

(a) Course Description
In this course unit, students shall understand the complete process of translating a program in a high-level language to machine language. The course gives an introduction to the design and implementation of a compiler with emphasis on principles and techniques for program analysis and translation. It also gives an overview of the tools for compiler construction. Topics covered include lexical and syntactic analysis, semantic analysis, type-checking, program analysis, code generation and optimization, memory management, and runtime organization.

(b) Course Objectives
The objectives of this course are to educate the students in:

- Expressing a programming language formally

- The analysis and synthesis of a compiler
- The different phases of a compiler
- How to optimize the machine code generated by the compiler to make it faster and more efficient
- The basic skills needed to design and implement a compiler of a given language
- Utilizing compiler construction tools in the process of building a compiler

(c) Learning Outcomes
By the end of the course, students should be able to:

- Describe the complete process of translating a program in a high level language to machine language
- Describe the design and implementation of a compiler
- Describe the principles and techniques for program analysis and translation
- Identify the tools for compiler construction

(e) Detailed Course Content

- **Module 1: Introduction (3 CH)**
  Why study compiler design, Language translators, Compilers versus interpreters, and the application of compilation techniques in other Computer Science fields. They will also have an overview of the phases of the compiler from analysis to synthesis phases.
- **Module 2: Lexical analysis (4 CH)**
  The lexical analysis of the compiler will be studied including lexical errors, scanner implementation, regular expressions, finite automata review (nondeterministic and deterministic FA, conversions), lexical analyzer generators.
- **Module 3: Syntax Analysis (10 CH)**
  Basics of syntax analysis, review of context free grammars, top-down parsers (recursive descent parsers, predictive parsers, LL parsers), FIRST and FOLLOW sets, parsing tables. Bottom-up parsers (LR parsers, Conflicts in LR grammars and how to resolve them).
- **Module 4: Semantic analysis (10 CH)**
  Basics of semantics analysis, semantic errors, syntax directed translations, formal semantics, attribute grammars (attributes and their computation),

type checking (static versus dynamic) and type inference, type equivalence (name versus structural), scope, inherited attributes and symbol tables. Overload resolution, polymorphism and dynamic dispatching

- **Module 5: Run-time organization and Intermediate code generation (4 CH)**
  Storage allocation(Static Versus Dynamic), Stack Allocation of Space, parameter passing mechanisms, heap management, garbage collection. Three-Address Code, control flow, register allocation.
- **Module 6: Code optimization (4 CH)**
  Classification, scope of optimization, machine independent versus machine dependent optimization, Principal Sources of Optimization, local versus global optimisation.
- **Module 7: Compiler Project (10 CH)**
  Hands on implementation group project of a simple compiler illustrating the different phases of the compilation process. There will be presentations of this project thereafter.

(d) Mode of Delivery
The course consists of a Blended learning including lectures, projects, and tutorials. The lecture component introduces the basic concepts of compiler writing. The project component will involve students in writing a compiler for a specified programming language.

(g) Mode of Assessment
Assessment will be by quizzes, assignments and/or tests (40%) and written examination (60%)

(h) Reading List

[1] Alfred V. Abo, Monica S. Lam, Ravi Sethi, and Jeffrey D. Ullman Compilers: Principles, Techniques, and Tools (2nd Edition) 2007.

[2] William .M. Waite and Gerhard Goos Compiler Construction. *Springer*, 1984.

[3] Kenneth C. Louden Compiler Construction: Principles and Practice 1st Edition. *Course Technology*, 1997

[4] Y.N. Srikant and Priti Shankar, The Compiler Design Handbook: Optimizations and Machine Code Generation, Second Edition, 2007 *CRC Press*, 2007.

[5] Keith D. Cooper and Linda Torczon Rice Engineering a Compiler, Second Edition, 2011

5.8.4 Computer Science Project 2

Course Name:	Computer Science Project 2
Course Code:	CSC 3211
Course Credit:	5
Contact Hours:	75
Year of Study:	3
Semester:	2

(a) Course Description
The Computer Science Project 2 is a continuation of Computer science project 1 which also aim at providing a platform for students to combine knowledge and skills acquired from Computer Science modules studied in the course in the production of a suitable Computer science project. The course is not a taught module but a major design and development exercise for the student carried out under supervision.

In the course, the students continue developing their project topics they selected in Computer Science Project 1, resulting into a substantial well written project report/dissertation and fully operational/working software solution or model that can be used by the end product users.

(b) Course Objectives

- To give experience of undertaking a significant individual/group design and development exercise from conception through to design, implementation and delivery.
- To allow students to conduct a substantial piece of research into unfamiliar area of computing.
- To introduce research methods and skills relevant for conducting such a project.
- To provide an opportunity for the students to synthesize what they have learned from other modules.

- To teach students planning, scientific writing and problem solving skills.
- To give students experience of working independently.
- To give students experience in writing up and presenting work in a scholarly fashion.

(c) Learning Outcomes
On successful completion of this course students will be able to.

- A general understanding of the state of the art in the student's chosen research area of specialization.
- Have an in-depth knowledge and understanding of the student's chosen area of specialization.
- have an understanding of the research methods required in conducting a research based project.
- Plan and carry out research appropriated to a project.
- Demonstrate project planning, scientific writing and problem solving skills.
- Manage personal learning and exercise critical self-evaluation.
- Have effective communication of general and specialized Computing Science concepts

(e) Detailed Course Content
Since the course is not a taught module but a major design and development exercise for the student carried out under supervision; and students are expected to select a research project from the topic of interest. The following a some the topics students can choose from

- Data Warehousing
- Internet of Things(IoT)
- Big Data/Data science
- Semantic Web
- Machine Learning
- Operating systems
- Multimedia computing
- Artificial Intelligence
- Data Mining
- Image Processing
- Bioinformatics
- Computer vision
- Algorithms and complexity

(d) Mode of Delivery
The course is a self-study/group-study or research exercise for the carried out under supervision.

(g) Mode of Assessment
A detailed project report/dissertation is submitted to the supervisor in the 14th week of the second semester for 40% (combined with student participation). A presentation of the results of the project is and well written report/dissertation are assessed by a panel of examiners for the remaining 60%

(h) Reading List

[1] Hossein Hassani. How to do the Final Year Projects. A Practical Guideline for Computer Science and IT Students.

[2] Ranjit Kumar. Research Methodology a step-by-step guide for beginners 3rd *Edition, 2014.*

[3] C.R. Kothari. Research Methodology methods and & techniques *Second Edition, 2004.*

5.8.5 Emerging Trends in Computer Science

Course Name:	Emerging Trends in Computer Science
Course Code:	CSC 3217
Course Credit:	3
Contact Hours:	45
Year of Study:	3
Semester:	2

(a) Course Description
The course is to expose provide students with an opportunity to search for knowledge in an area of interest. It is to allow a student do lightweight research and explore the current trends in a certain computer science area.

(b) Course Objectives
The objectives of the course are to:

1. aid a student get an in depth understanding of the developments in at least one area of computer science
2. improve the students research skills
3. develop confidence in the students on the ability to search for knowledge with little guidance

4. improve students ability to present and report findings from self study on new technologies and trends

(c) Learning Outcomes
By the end of the course, the learners should be able to

1. Discuss the various areas in computer science that are potential areas of research
2. Demonstrate skills acquired from topics of self study.

(e) Detailed Course Content
The content is not specific but will be dependent on the area the student chooses to pursue.

(f) Study Materials
The materials shall include technology magazines, conference proceedings, technology talks and shows, online resources and case studies.

(d) Mode of Delivery
The study will be blended learning including a mix of student centered and learning teacher centered pedagogy. Students will be grouped in theme areas which will be covered by a set of staff. Staff will guide students on the specific themes/topics in the area as well as where to search for information. Staff will also address experiences and hardships found.

(g) Mode of Assessment
Assessment will include

1. Class presentations and take home projects will constitute progressive assessment (40%)
2. Final examination that will constitute 60%. The examination will include student presentation, reports, demonstrations about the selected study area.

(h) Reading List

[1] This will be decided by the lecturer in consultation with the Course leader and dependent on the selected study area/topic.

5.8.6 Distributed Systems Development

Course Name:	Distributed Systems Development
Course Code:	BSE 3202
Course Credit:	4
Contact Hours:	60
Year of Study:	2
Semester:	1

(a) Course Description

This course gives students theoretical and practical skills on development of distributed systems and applications. This includes distributed-system specific challenges like interprocess communication, reliability and robustness. Protocols like Remote Mothod Invocation and Remote Procedure Calls should be covered using languages like Java and C respectively.

(b) Course Objectives

The aim of the course is to equip students with skills of developing distributed systems.

(c) Learning Outcomes

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

- describe Event driven Architectures
- describe the development, documentation and testing of distributed applications
- describe the techniques of reusable, extensible and efficient software systems
- discuss maintainability and concurrence in distributed systems
- describe abstraction based on patterns and object-oriented techniques

(e) Detailed Course Content

- **Module 1 (16 hours): Distributed Objects**
  - Event-driven software architectures
  - Distributed object computing (RPC and RMI)
- **Module 2 (24 hours): Distributed Applications**
  - Development, documentation and testing of distributed applications
  - Techniques for reusable, extensible and efficient software systems
  - Practical project(s)
- **Module 3 (20 hours): Advanced techniques**
  - Maintainability and concurrence in distributed systems
  - Information Security in Network Based Applications

(d) Mode of Delivery

Teaching will be by blended learning including lectures and laboratory demonstrations

(g) Mode of Assessment

- To assess whether students have met the learning outcomes, there will be course work that will contribute 40% to final grade and a final exam that will constitute 60% to the final grade.
- The coursework will comprise at least one (practical) assignment and at least one (written) test.
- The Final exam will be done at the end of the semester and will take a duration of three (3) hours.

(h) Reading List

1. Distributed Systems Concepts and Design by George Coulouris, Jean Dollimore, Tim Kindberg and Gordon Blair, Fifth Edition by Addison Wesley, May 2011, ISBN: 0-13-214301-1
2. S. Tanenbaum and M. V. Steen, Distributed Systems: Principles and Paradigms, Second Edition, Prentice Hall, 2006, ISBN: 0132392275

3. G. Coulouris, J. Dollimore, and T. Kindberg, Distributed Systems: Concepts and Design, 3rd Edition, Addison-Wesley, 2000, ISBN: 0201619180
4. R. Anderson, Security Engineering: A Guide to Building Dependable Distributed Systems, John Wiley & Sons, 2001, ISBN: 0471389226

5.8.7 Data Warehousing and Business Intelligence

Course Name:	Data Warehousing and Business Intelligence
Course Code:	IST 3109
Course Credit:	4
Contact Hours:	60
Year of Study:	3
Semester:	2

(a) Course Description

This course provides candidates with practical and theoretical skills in data and information management technologies so as to provide decision support capabilities. The course provides skills on building of data warehouses, managing and querying data warehouses and utilization of business intelligence for decision-making. Modeling techniques for futuristic prediction is also covered.

(b) Course Objectives
The aims of the course are to:

1. Make students understand the role and operation of data warehouses
2. Equip students with skills of developing data warehouses
3. Equip students with skills of maintaining existing data warehouses
4. Equip students with skills of manipulating data warehouses to generate information for business decision making

(c) Learning Outcomes

After successfully completing this course, students should be able to:

1. Explain how data warehousing combined with good business intelligence can increase a company's bottom line and the different forms of business intelligence. Distinguish between on-line transaction processing (OLTP)

and online analytical processing (OLAP); describe how structured, semistructured, and unstructured data are all essential elements of enterprise information and knowledge management.

2. Create a simple data warehouse ("data mart") and develop dimensional models from which key data for critical decision-making can be extracted.
3. Load extracted and transformed data into the data warehouse.
4. Understand the different kinds of data mining algorithms and sketch out the process for extracting data from disparate databases and data sources.

(e) Detailed Course Content

- Module 1 (16 hours): **Basic concepts**
  - Data warehouse concepts: Partitioning, Granularity, Record of source, and Metadata
  - Building viable decision support environments
- Module 2 (24 hours): **Data manipulation**
  - Architect development
  - Use of operational data stores, and transactional systems
  - Data migration and integration
- Module 3 (20 hours): **Online analytics**
  - Online analytic processing
  - ROLAP vs MOLAP
  - Data mining

(d) Mode of Delivery

There will be blended learning including Lectures, Lab practicals, Online learning management systems, Class discussions and presentations, and Group project-based assignments

(g) Mode of Assessment

The assessment will be done by tests (20%), take home practical assignments (20%), and a final written examination (60%)

(h) Reading List

1. Ponniah, P. (2011). Data warehousing fundamentals for IT Professionals, 2nd Edition. Publisher: John Wiley & Sons.

2. Haertzen, D. (2012). The Analytical Puzzle: Profitable Data Warehousing, Business Intelligence and Analytics. Publisher: Technics Publications Llc.

3. Khan, A. (2011). Business Intelligence and Data Warehousing Simplified: 500 Questions, Answers, & Tips. Publisher: International Pub Marketing.

4. Taniar, D. (2009). Progressive Methods in Data Warehousing and Business Intelligence: Concepts and Competitive Analytics. Illustrated Edition. Publisher: IGI Global Snippet.

5. Roland Bouman, R. (2009) Pentaho Solutions: Business Intelligence and Data Warehousing with Pentaho and MySQL.

6 Resources and Infrastructure

6.1 Academic Staff

The School of Computing and Informatics Technology has staff who can competently teach the course units of the programme. The list of staff members in the Department of Computer Science and other departments is in Appendix A.

6.2 Lecture Space

The School of Computing and Informatics Technology is housed in the College of Computing and Information Sciences with 2,500 and 12,000 square meter buildings known as block A and B, respectively. Block A mainly accommodates offices for the administration and teaching staff. The ground floor has a video conferencing facility; on second floor there are two teaching labs, a lecture room and two seminar rooms used for teaching purposes as well. On fourth floor, there is a seminar room and a conference hall and the sixth floor has a testing centre. Block B has lecture rooms together with the rest of the general and specialized laboratories i.e. Software Systems Centre, Multimedia lab, Mobile Applications among others. The two buildings sufficiently cater for all the lecture and laboratory space requirements at the College of Computing and Information Sciences (CoCIS). Specifically, CoCIS has twelve lecture theaters each of 200 square meters (300 seat capacity) of circulation space where students are able to access other services such wireless Internet services.

6.3 Computer Laboratories and Software

The CoCIS buildings that house CIT, i.e. Block A and B, have general computing laboratories (for student hands-on training); teaching and specialized laboratories, that are shared amongst the four departments. The School has 7 laboratories each of 800 square meters (1000 seat capacity) and six small laboratories of total area 1200 square meters with a total of approximately 700 computers. All computers in the laboratories are pre-installed with various operating systems and computing applications with a focus on open source applications. The School has access to software for the practical aspects of the programme.

The School of Computing and IT has also put in place specialized research laboratories (e.g. the Multimedia Laboratory, Geographical Information Systems Laboratory, Mobile Computing Laboratory, Networking and Systems Laboratory, Software Incubation Laboratory, Computer Engineering Laboratory and e-learning Laboratory) and plans are under way to establish more laboratories using funds available under donor funded projects and internally generated funds.

6.4 E-Learning Platforms

Makerere University has an eLearning platform known as Muele (`http://www.muele.mak.ac.ug`) and it is expected that courses will be developed as interactive online modules on Muele. Students in the Department of Computer Science have adequate access to computers. Each student will be expected to have a personal computer. This creates a good environment for e-learning blended teaching. All courses in the new curriculum will be taught in a blended way. All course materials will be put on Muele. Staff will, as much as possible, make use of e-learning facilities like discussion forum and drop boxes for assignments. This will increase student activity/participation and reduce staff effort (e.g. staff will not need to dictate notes). This in turn will increase the material covered and taken in by the students.

6.5 Library Services

Makerere University library supports the College of Computing and Information Science library, which is located on the first level of Block B. The College Library is stocked with up-to-date information resources. The information resources in the College Library have been acquired through purchases made by Makerere University Library and the College of Computing & Information Sciences. Additionally, the University Library has dedicated space for graduate students and provides access to print books, print journals, electronic journal databases, a well-stocked reference section and connections to many online databases like the Uganda Scholarly Digital Library at `http://dspace3.mak.ac.ug`. The print collection is beefed up by the broad variety of electronic resources provided by the University Library and accessible online at `http://muklib.mak.ac.ug`. Through the document delivery service, users who fail to get access to full-text articles from the available databases can make requests for articles, which are delivered, to them at no cost. Library users can also access the Online Public Access Catalogue (OPAC) to get bibliographic information about the collections found in the College Library. Below is a list of all electronic databases that Makerere subscribes to;

1. Institute of Electrical and Electronic Engineers (IEEE)
2. Emerald Insight
3. Springer Verlag
4. Research4life (ARDI & HINARI)
5. Sage Publications
6. E-library (eBook database)
7. Science Direct
8. Association of Computing Machinery (ACM) Digital Library

7 Quality Assurance

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

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

(b) Ascertain the Implementation of the methodological changes proposed

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

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

7.1 Feedback from Students Enrolled

In the current set up, the masters class has a student representative who is in constant contact with the Head of Department in case there are any quality related matters in a particular class. This set up is to be maintained.

At the end of the semester, samples of students are given questionnaires to respond to several quality related matters like staff punctuality, delivery mode, course content and the general perceived usefulness of the course unit. The School of Computing and Informatics Technology is the process of creating a computerized system that will capture and analyze the data. With the computerized system:

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

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

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

7.2 Class meetings

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

7.3 Peer review

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

7.4 External examiners' reports

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

7.5 Tracer studies

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

8 Appendix A: Academic Staff List

Name	Qualifications and Awarding Institution	Area of Specialization	Current Load per Week (hrs)	Proposed New Load (hrs)	Staff Status
Assoc. Prof. Engineer Bainomugisha	PhD CS (VUB) [1] MSc. CS (VUB) BSc. CS (Mak)[2]	Programming Language Engineering Cloud & Mobile Computing Secure Programming Distributed Systems Crowd-sourcing/Citizen Science	12	12	Full time (Chair)
Dr. John Ngubiri	PhD. CS (RUN)[3] MSc. CS (Mak) PGD. CS (Mak) BSc. Educ (Mak)	Software Security Parallel Systems Process Modelling Algorithms	12	12	Full time
Dr. Joyce Nakatumba-Nabende	PhD. CS (TUE)[5] MSc. CS (Mak) BCS (MUST)	Data & Process Mining Business Process Management Machine Learning	12	12	Full time
Dr. Daudi Jjingo	PhD. Bioinformatics (Georgia) MSc. Bioinformatics (Leeds) BSc. Biochemistry (Mak)	Bioinformatics of Infectious and Chronic Diseases Health Informatics	9	12	Full time
Dr. Peter Nabende	PhD. Math & Natural Sciences (RUG) MSc. CS (Mak) BSc. Mech Eng (Mak)	Intelligent Systems Computational Linguistics Data Mining	9	9	Full time
Dr. Florence Nameere Kivunike	PhD. Computer Systems Sciences (Stockholm) MSc. CS (Mak) BSc Elec Eng (Mak)	E-service Delivery Web systems Development	9	12	Full time
Dr. John Quinn	PhD. CS (ED)[6] BA. CS (CM)[7]	Data Science Computer Vision Artificial Intelligence	-	-	Honorary Assoc prof
Ms. Barbara Nansamba	MSc. DCSE (Mak)[2] BSc. CS (Mak)[2] Computer Networks	Data Science Computer Vision Computational Maths	12	12	Full time

Name	**Qualifications and Awarding Institution**	**Area of Specialization**	**Current Load per Week (hrs)**	**Proposed New Load (hrs)**	**Staff Status**
Mr. Dragule Swaib	MSc. CS [8] BSc. CS (IUIU)[8] Software Engineering	Robotics Data Structures Compiler Design	-	12	Study leave
Mr. Emmanuel Lule	MSc. DCSE (Mak) [2] BSc. CS (IUIU)[8] Programming	Networks Operating Systems Embedded Systems	-	12	Study leave
Mr. Jonathan Kizito	MSc. CS () [2] BSc. (Mak)[2] Mathematics	Distributed Systems Programming Software Engineering	12	12	Full time
Dr. Rose Nakibuule	MSc. CS & BSc. (Mak) [2] BSc. CS(Dar)[0] Logic Programming	Mathematics Artificial Intelligence Computer Vision	12	12	Full time
Ms. Marriette Katarahweire	MSc. CS (VUB) [1] BSc. CS(Mak)[2] Compiler Design	Computer Security Programming Mobile Computing	6	12	Study leave
Dr. Michael Kizito	MSc. CS; PhD [9] BSc. CS(Mak)[2] Networks	Digital Innovation IT Management Software Engineering	-	12	Full Time

[1] Vrije Universiteit Brussels, Belgium
[2] Makerere University, Uganda
[3] Radbound University Nijmegen, The Netherlands
[4] Rijks Universiteit Groningen, The Netherlands
[5] Technische Universiteit Eindhoven, The Netherlands
[6] University of Edinburgh, UK
[7] University of Cambridge, UK

[8] IUIU, Uganda

[9] Goteborg University, Sweden

9 Appendix B: Projected Revenue & Expenditure

SN	Item	Rate	Amount (UGX)
A	**Revenue collection**		
A.1	Tuition fees from Ugandan students	UGX 3,024,000 x 200 students	604,800,000
A.2	Tuition fees from International students	UGX 4,536,000 x 20 students	90,720,000
		Total Revenue	**695,520,000**
B	**Expenditure**		
B.1	Teaching Expenses	38%	264,297,600
B.2	ICT/specialized equipment & research	20%	139,104,000
B.3	Staff Development	5%	34,776,000
B.4	Administrative Expenses	3%	20,865,600
B.5	Office Expenses	3%	20,865,600
B.6	Library Materials	5%	34,776,000
B.7	Utilities/Furniture	1%	6,955,200
B.8	University Council	25%	173,880,000
		Total Expenditure	**695,520,000**

10 Appendix C: Previous Curriculum

Key

1. **LH**: Means Lecture Hour.
2. **PH**: Means Practical Hour.
3. **CH**: Means Contact Hour.
4. **TH**: Means Tutorial Hour.
5. **CU**: Means Credit Unit.

10.0.1 Year I, Semester I

Code	**Course Name**	**LH**	**PH**	**CH**	**TH**	**CU**
CSC 1100	Computer Literacy	30	60	-	60	4
BIS 1104	Communication Skills for IT	45	30	-	60	4
CSC 1104	Computer Organization & Architecture	60	-	30	45	4
CSC 1108	Individual Project 1	15	90	-	60	4
CSC 1107	Structured Programming	30	30	-	45	3
TOTAL						**19**

10.0.2 Year I, Semester II

Code	Course Name	LH	PH	CH	TH	CU
BIS 1206	Systems Analysis and Design	45	-	30	60	4
MTH 1203	Calculus 1	45	-	30	60	4
CSC 1214	Object Oriented Programming	30	60	-	60	4
MTH 2203	Numerical Analysis 1	45	-	30	45	3
BIS 1204	Data & Information Management 1	30	60	-	60	4
TOTAL						**19**

10.0.3 Year I, Recess Term

Code	Course Name	LH	PH	CH	TH	CU
CSC 1304	Practical Skills Development	15	90	-	75	5
CSC 1303	Cisco Certified Network Associate (CCNA) **Audited**	150	-	-	75	5
TOTAL						**5**

10.0.4 Year II, Semester I

Code	**Course Name**	**LH**	**PH**	**CH**	**TH**	**CU**
CSC 2114	Artificial Intelligence	30	30	-	60	4
BSE 2103	Computer Networks	45	30	-	60	4
MTH 3105	Discrete Mathematics	30	-	30	45	3
CSC 2118	Data Structures & Algorithms	45	-	30	60	4
Electives (select 1)						
CSC 2208	Formal Methods	45	-	30	60	4
CSC 2207	Software Engineering	45	-	30	60	4
TOTAL						**18**

10.0.5 Year II, Semester II

Code	**Course Name**	**LH**	**PH**	**CH**	**CU**	Remark
CSC 2200	Operating Systems	45	-	30	60	4
CSC 1209	Logic Programming	30	30	-	45	3
CSC 2218	Systems Programming	45	-	30	60	4
CSC 2210	Automata, Complexity and Compatibility	45	-	-	45	3
BIT 2207	Research Methodology	30	-	30	45	3
TOTAL						**17**

10.0.6 Year II, Recess Term

Code	**Course Name**	**LH**	**PH**	**CH**	**CU**	Remark
CSC 2303	Field Attachment	-	300	75	5	Modified
TOTAL					**5**	

10.0.7 Year III, Semester I

Code	**Course Name**	**LH**	**PH**	**CH**	**TH**	**CU**
CSC 3110	User Interface Design	45	30	-	60	4
BAM 2102	Entrepreneurship Principles	30	-	30	45	3
CSC 3112	Principles of Programming Languages	45	-	-	45	3
CSC 3118	Computer Science Project I	-	-	150	75	5
	Electives (select 1)					
CSC 3121	Computer Graphics	30	30	-	45	3
CSC 3115	Advanced Programming	45	30	-	45	4
BIS 3100	Modelling and Simulation	30	30	-	45	3
MTH 3107	Linear Programming	30	-	-	30	3
TOTAL						**18**

10.0.8 Year III, Semester II

Code	**Course Name**	**LH**	**PH**	**CH**	**CU**	Remark
BSE 2206	Data Communications	45	30	-	60	4
CSC 3205	Compiler Design	45	30	-	45	3
CSCS 3211	Computer Science Project II	-	-	150	75	5
	Electives (Select 1)					
CSC 3207	Computer Security	45	30	-	45	3
BSE 3202	Distributed Systems Development	45	30	-	60	3
BIS 3205	Data Warehousing and Business Intelligence	45	30	-	60	3
CSC 3217	Emerging Trends in Computer Science	45	45	-	45	3
TOTAL					**19**	