MAKERERE UNIVERSITY

COLLEGE OF COMPUTING AND INFORMATION SCIENCES

SCHOOL OF COMPUTING AND INFORMATICS TECHNOLOGY

DEPARTMENT OF NETWORKS

P.O. BOX 7062, KAMPALA, UGANDA

A REVISED BACHELOR OF SCIENCE IN SOFTWARE ENGINEERING (BSSE)

FEBRUARY 2021

Contents

1.0 Introduction

The ICT industry continues to change not only in Uganda, but Globally. Mobile communications technologies are booming due to the high demand for ubiquitous services. To harness the benefits of mobile infrastructures, we require having local content and service developers. The rise of the "Internet of Things (IoT)" concept in recent year has gained enormous momentum, and is now one of the most popular topics in technology. It is projected that there will be approximately 26 Billion connected devices by 2020, meaning programmes need to consider this new paradigm. Makerere University and CoCIS missions emphasize world class teaching and research that is relevant not only for the region, but globally.

The programme emphasizes all aspects of learning, i.e., theory, research, practical and hands-on skills, transferable skills, group working, and so forth with a view to build a world-class and competitive research-oriented programme. The goal is to fulfil the needs of the booming ICT industry such as software innovative services, as well as the needs for skilled researchers in software engineering.

Whereas the previous curriculum of BSc. Software Engineering had a mix of both softeware enginnering and computer networks, this revision of the curriculum focused more on producing a professional software engineer ready to meet the current demands of software industry. The revisions in this curricullum have been based on several concerns observed during the implementation of the previous curriculum (2009/2010 - 2018/2019) among others including: the need to improve and increase software engineering research capacity, and productivity leading to innovation development in Uganda; as Makerere University continues striving to become a research-led University.

1.1 Background

1.1.1 College of Computing and Information Sciences (CoCIS)

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.

CoCIS Vision: To be a leader in computing and information sciences education, research and services internationally.

CoCIS Mission: To provide first class teaching, research and services in computing and information sciences responsive to national and international needs.

1.1.2 School of Computing and Informatics Technology (SCIT)

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.

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

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

SCIT 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.

1.1.3 Department of Networks

The Department of Networks is responsible for running programmes in the following areas: Data Communications Engineering; Software Engineering; Hardware Engineering; Network and System Administration; Network Planning, Design and Management; Data Communication Networks; ICT Policy and Regulation; and related areas. The Department currently runs the following programmes approved by National Council for Higher Education through the University Council:

(i) PhD in Software Engineering;

(ii) Master of Science in Data Communications and Software Engineering (MDSE) (Communication Networks and Mobile Technologies option; and Software Engineering option);

(iii) Postgraduate Diploma in Data Communications and Software Engineering (PDSE) (Communication Networks and Mobile Technologies option; and Software Engineering option);

(iv) BSc. Software Engineering (BSSE)

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

(i) Networked Systems and Systems Security (NetS);

(ii) The Software Systems Center (formerly Software Engineering Research Group);

1.2 Bachelor of Science in Software Engineering (BSSE) Programme

This programme was first launched in August 2009. It offers a course of study leading to the BSc. in Software Engineering. The curriculum document was last revised in the year 2013 which is clearly over due for revision.

1.2.1 Who is a Software Engineering Graduate?

A Software Engineering Graduate has an adequate grasp of the required principles and techniques to produce software systems on time, within budget and with few or no defects. Our graduate is expected to apply these principles of engineering to the design, development, maintaining, testing, and evaluation of the software and systems that make computers or anything containing software work. Thus, the training given in our program ensures that our graduates have a personal, business and technical skill to advance their career wherever they want to go. In particular, we prepare them to understand that choosing an occupation, getting that first graduate job, and growing in that job will require planning, self awareness, flexibility and a forward-looking attitude to help manage the personal transitions, as well as the technological and economic changes of the future. In that regard, BSSE program offers diverse career options as including:

i. Software developer/ engineer,
ii. Software architect,
iii. Analyst/ programmer,
iv. Games developer,
v. Project manager,
vi. Mobile application developer,
vii. Test analyst/engineer

1.3 Justification for Revision of the B.Sc. Software Engineering Programme

Uganda's Vision 2040 fronts ICT as one of the key drivers for national development. Software industry in Uganda has been registering a steady growth and the Government of Uganda through the National Information Technology Authority(NITA) and Buy Uganda Build Uganda (BUBU) is advocating for locally developed software as a way of support local software engineers. The BSc. Software Engineering Programme was the country's pioneer software engineering course in 2009 and has over the years been producing software engineers for the country and beyond.

The BSSE programme was last revised in 2013. National Council for Higher Education (NCHE) recommends revision of academic programmes every after five years. In addition, with the continuously changing Information and Communication Technology field and an economy that is based on dynamic skills, we realized a need for revising the curriculum to update the programme and 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:

i. Requirement from NCHE to review academic programmes every 5 years.
ii. A review of the trends in software industry which did not exist at the time of the past curriculum revision. This was done via faculty discussions, particularly the need to: a)focus on research in strategic areas of importance to the regional and international needs; b) equip students with skills and techniques to carry out high quality research.
iii. Market analysis: We undertook a preliminary market analysis to evaluate the attractiveness and the dynamics of software engineering market within East Africa with special attention to the kind of jobs/roles being advertised within this market segment and the different skill sets that are being sought. We compared this with the internship feedback obtained from our industrial

partners where our students get placements. With this information, we were able to identify the opportunities, strengths, weaknesses and threats of the programme and necessary adjustments.

iv. To remove redundancy that existed in the previous curriculum e.g. reduced from three knowledge areas (tracks/options) to two focused knowledge areas.

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

vi. Benchmarking similar programmes in professional bodies (e.g. Association of Computing Machinery) and recognised universities regionally and internationally but also to meet the current changes in software standards that meet the development plans at the nation, region and global level.

vii. To expand the depth and breadth of the content for a complete BSSE graduate.

viii. External examiners' reports: These reports were also a great input into this review process. Although most provided comments were majorly on assuring quality of students' learning experiences and fair assessment, some comments were intended to iimprove the program and therefore significant for this revision.

1.4 Programme Objectives

The general objective of the programme is to train/skill student the theory and practices of; design, development, securing and maintenance of software.

Other objectives of the BSSE programme are:

- To build human resource capacity that is proficient in software design and development in a variety of languages and platforms using a methodical approach for both the public and private sectors. Such human resource that is well grounded in the fundamental concepts of software engineering and able to continue their professional development throughout their careers.
- To equip graduates with lifelong learning skills capable of functioning responsibly in diverse environments and able to work in teams.
- To empower graduates with ability of creating their own jobs in software engineering field leading to the fourth industrial revolution.
- To foster interdisciplinary studies which enable the application of software knowledge in non-computating disciplines.

1.5 Programme Outcomes

- Increased human resource capacity to design and develop resilient software solutions for the industry.
- Software engineering research collaborations and partnerships.
- Surficient employment for the graduates in the industry
- Advancement in multidisciplinary fields in connection with software engineering

1.6 Career Options arising from the BSSE Programme

Software engineering graduates find themselves in a variety of environments in academia, research, industry, government, private and business organizations – analyzing problems for solutions, formulating and testing systems, or working in teams for product development. Here's a short list of research and vocational areas for BSSE graduates:

- Network Systems Engineering – Network Design, Monitoring and maintenance of net- work equipment, Notification of all network outages to customers, Troubleshooting and configuration of network device
- Software Engineering – Develop methods for the production of software systems on time, within budget, and with few or no defects.
- Computer Theory – Investigate the fundamental theories of how computers solve problems, and apply the results to other areas of computer science.
- Operating Systems and Networks – Develop the basic software used by computers to supervise themselves or to communicate with other computers.
- Software Applications – Apply computing and technology to solving problems outside the computer field - in education or medicine, for example.
- Computer Design and Engineering – Design new computer circuits, microchips, and other electronic components.
- Computer Architecture – Design new computer instruction sets, and combine electronic or optical components to provide powerful but cost-effective computing.

The careers are described in IEEE Computer Society[1] and Computing; ISBN 185824 489 7; Quality Assurance Agency for Higher Education 2000; Published by Quality Assurance Agency for Higher Education, Southgate House, Southgate street, Gloucester GL1 1UB, Tel 01452 557000, Fax 01452 557070, web www.qaa.ac.uk; Printed by Kall Kwik, Gloucester.

1.7 The review process

The review process was sparked by multiple factors highlighted in section 1.3. The major purpose of the process was to ensure that the program is operating on the current standards in the area of software engineering and aligned to the general guidelines for degree programmes in Uganda. Emphasis was put on Uganda's development plan and a need to train graduates who can accomplish these plans. Among the major processes included; consultations with the industry, examiner's reports, the student's body including alumni of the programme, staff in the college concerned with the programme.

1.8 Summary Table for review Composition of Old, Modified and New Course Units

The table below shows the percentage composition of the old, new and modified course units in the revised curriculum. Old means that the course unit was retained from the old curriculum without any changes, Modified means that the course unit of the old curriculum was modified to accommodate advances in the area while New means that the course unit is completely new in the revised curriculum.

[1] www.computer.org/education/careers.htm

Year	Semester	Old	Modified	New	Total
Year I	Semester I	3	1	2	**6**
	Semester II	3	0	3	**6**
Year II	Semester I	3	1	1	**5**
	Semester II	2	2	2	**6**
Year III	Semester I	3	1	1	**5**
	Semester II	3	0	4	**7**
Year IV	Semester I	1	1	2	**4**
	Semester II	1	1	2	**4**
Total		**20(46.5%)**	**7(16.3%)**	**16(37.2%)**	**43(100%)**

1.9 Statement to justify existence of BSSE as a standalone programme.

The view of the school board, College and other professionals in the ICT Research and Practice sector is that the BSSE should stand alone. The position is premised on the following grounds:

i. In the current technological world, CS and SE professions have been distinct for the last 50 years. They lead to different STEM degrees. With the current trends, they are further getting apart not converging.
ii. The national ICT policy for Uganda (2014) calls for the Enhancement of research and innovation in ICT products, applications, and services. Specifically, it prioritizes the promotion of software and applications development. Both professionals are core to technological development and by extension national development and as such Uganda will lose out if one is halted. The proposed merger creates a contradiction for this policy (see attachment).
iii. An attempt to produce both in a single curriculum will lead to producing half-baked graduates of each other and it will be a bigger disaster.
iv. Makerere University with over 40 PhDs in CS/SE area has a competitive advantage in Uganda and the East African region. Terminating the program internally will be an own goal, shooting ourselves in the foot.
v. The nurturing ground for researchers and innovators at advanced degree level will be destroyed – other universities have less capacity.

A detailed technical analysis of the justification was submitted to council with all comparisons and approved to avert the merger of the two programmes. This can be erefered to anytime.

1.10 Controversial Comparison of the BSSE with the B.S. Computer Science

Bachelor of Science in Computer Science								Revised BSSE							
Year 1 Semester 1								**Year 1 Semester 1**							
Code	Name	LH	PH	CH	CU			Code	Name	LH	PH	TH	CH	CU	CHANGES
CSC 1102	Structured & Object-Oriented Programming	45	30	60	4			BSE1106	Problem Solving and Programming Concepts	4	30	30	30	60	Modified
CSK 1101	Communication Skills	30	60	60	4			UNV1101	Communication Skills	4	30	60		60	Old
CSC 1105	Mathematics for Computer Science	30	60	60	4			BSE1107	Mathematics for Software Engineers	3	30	-	30	45	New
CSC 1104	Computer Architecture & Organisation	30	60	60	4			BSE1108	Technical Analysis and Design	4	30	30	30	60	New
CSC 1109	Digital Innovation & Computational Thinking	30	60	60	4			IST1101	Foundations of Information Systems & Technology	4	45	30	-	60	Old
	Audit Course								Audit Course						
CSC 1100	Computer Literacy	30	60	60	4			CSC1109	Computer Literacy	4	30	30	30	60	Old
	Total CU				**19**				**Toal CU**	**19**					
Year 1 Semester 2								**Year 1 Semester 2**							
CSC 1200	Operating Systems	30	60	60	4			BSE1206	Software Development Principles	3	45	-	-	45	Old
CSC 1201	Probability & Statistics	30	60	60	4			MTH2203	Numerical Analysis I	3	30	-	30	45	Old
CSC 1202	Software Development Project	15	90	60	4			BSE1209	Object Oriented Programming I	4	30	30	30	60	New
IST 1204	Systems Analysis and Design	30	60	60	4			IST1203	Data and Information Management I	4	30	60	-	60	Old
CSC 1204	Data Structures and Algorithms	30	60	60	4			BSE1208	Introduction to Web Development	4	30	30	30	60	New
	Total CU				**20**				**Total CU**	**18**					
Year 1 Recess Term								**Year 1 Recess Term**							
CSC 1303	Cisco Certified Network Associate (CCNA) Audited														
CSC 1304	Practical Skills Development	15	90	75	5			BSE 1302	Software Engineering Practical Skills Project I	5	-	150	-	75	Old
Year 2 Sem 1								**Year 2 Sem 1**							

BSE 2103	Computer Networks	30	60	60	4		CSC2114	Artificial Intelligence	4	30	30	30	60	Modified
CSC 2107	Database Management Systems	30	60	60	4		CSC2100	Data Structures and Algorithms	4	30	30	30	60	Old
CSC 2114	Artificial Intelligence	30	60	60	4		BSE2106	Computer Networks	4	30	30	30	60	Old
CSC 2118	Embedded and Real-time Systems	30	60	60	4		BSE2105	Formal Methods	3	45	-	-	45	Old
CSC 3105	Discrete Mathematics	30	30	45	3		BSE2107	Object Oriented Programming II	4	30	30	30	60	New
	Total CU				**19**			**Total CU**	**19**					
Year 2 Sem 2							**Year 2 Sem 2**							
IST 2203	Research Methodology	30	60	60	4		CSC2200	Operating Systems	4	30	30	30	60	Modified
CSC 2206	Introduction to Machine Learning	30	60	60	4		BSE2207	Emerging Web Development Technologies	4	30	30	30	60	New
CSC 2209	Cloud Computing	30	60	60	4		BSE2208	Requirements Engineering	3	45	-	-	45	Modified
CSC 2210	Automata, Complexity and Compatibility	30	30	45	3		BSE2209	Mobile Programming Project	4	15	60	30	60	New
							BSE2206	Data Communication	4	45	30	30	60	Old
	Electives													
CSC 2207	Robotics	30	60	60	4									
CSC 2208	Software Quality and Verification	30	60	60	4									
CSC 2218	Software Construction	30	60	60	4									
	Total CU				**19**			**Total CU**	**19**					
Year 2 Recess Term							**Year 2 Recess Term**							
CSC 2303	Field Attachment	-	300	75	5		BSE2302	Software Engineering Practical Skills Project II	5	-	150	-	75	Old
Year 3 Sem 1							**Year 3 Sem 1**							
BAM 2102	Entrepreneurship Principles	30	30	45	3		BSE3110	Object Oriented Analysis and Design	4	30	30	30	60	Old
CSC 3115	Advanced Programming	30	60	60	4		BSE3111	Embedded Systems I	4	30	30	30	60	New

CSC 3118	Computer Science Project I	-	150	75	5		BSE3104	Software Metrics	3	45	-	-	45	Old
CSC 3110	User Interface Design	45	30	60	4		CSC3110	User Interface Design	4	30	30	30	60	Modified
	Electives							**Electives**						
IST 3110	Business Process Management	30	60	60	4		BSE3106	Mobile Networks and Computing	4	45	30	-	60	old
CSC 3114	Cryptology and Coding Theory	45	-	45	3		BSE3105	Software Evolution	4	45	-	30	60	Old
CSC 3117	Operations Research	30	30	45	3									
CSC 3121	Computer Graphics	30	30	45	3									
IST 3208	Modelling and Simulation	30	30	45	3									
	Total CU				**19/20**			**Total CU**	**18**					
Year 3 Sem 2							**Year 3 Sem 2**							
CSC 3205	Compiler Design	30	30	45	3		BSE3210	Software Architecture and Patterns	3	45	-	-	45	New
CSC 3207	Computer Security	30	30	45	3		BSE3211	Software Testing and Verification	3	45	-	-	45	New
CSCS 3211	Computer Science Project II	-	150	75	3		IST2203	Research Methodology	4	45	-	30	60	Old
CSC 3217	Emerging Trends in Computer Science	30	30	45	3		CSC2206	Machine Learning	4	30	30	30	60	Old
	Electives							Electives						
BSE 2206	Data Communications	30	60	60	4		BSE3214	Cloud Computing and Big Data	4	30	30	30	60	New
BSE 3202	Distributed Systems Development	30	60	60	4		BSE3213	Embedded Systems II	4	30	30	30	60	New
BIS 3205	Data Warehousing and Business Intelligence	30	60	60	4									
	Total CU				**18**			**Total CU**	**18**					
							Year 3 Recess Term							
							BSE 3302	Field Attachment	5	-	150	-	75	Old
							Year 4 Sem 1							
							BSE4100	Software Engineering Project I	5	-	150	-	75	Old
							BSE4102	ICT Innovation and Entrepreneurship	3	45	-	-	45	New

								BSE4104	Emerging Trends in Software Engineering	3	45	-	-	45	Modified
								BSE4105	Software Integration and Deployment	4	30	60	-	60	New
									Total CU	**15**					
								Year 4 Sem 2							
								BSE4200	Software Engineering Project II	5	-	150	-	75	Old
								BSE4202	Software Security	4	30	60	-	60	Modified
								BSE4203	Software Engineering Standards and Ethics	3	45	-	-	45	New
								BSE4204	Software Quality Management	3	45	-	-	45	New
									Total CU	**15**					
	Total courses = 42								**Total courses = 44**						
	Similar courses = 8/42 ~ 19.0 %								**Similar courses = 8/44 ~ 18.2%**						

There is approximately 18.5% similarity in the revise content of the two curriculum. The similarity is due to the shared computing foundations and the university cross cutting courses.

1.10 Comparison of current and the revised curriculum.of BSSE

Current BSSE								Revised BSSE							
Year 1 Semester 1								**Year 1 Semester 1**							
Coures code	Name	LH	PH	TH	CH	CU		Code	Name	LH	PH	TH	CH	CU	CHANGES
BSE 1106	Problem Solving and Programming Concepts	30	30	-	45	3		BSE1106	Problem Solving and Programming Concepts	4	30	30	30	60	Modified
BIS 1104	Communication Skills for IT	30	60	-	60	4		UNV1101	Communication Skills	4	30	60		60	Code updated
CSC 1100	Computer Literacy	30	60	-	60	4		BSE1107	Mathematics for Software Engineers	3	30	-	30	45	New

CSC 1107	Structured Programming	30	30	-	45	3	BSE1108	Technical Analysis and Design	4	30	30	30	60	New
BIS 1100	Foundations of Information Systems	45	-	-	45	3	IST1101	Foundations of Information Systems & Technology	4	45	30	-	60	Old
								Audit Course						
							CSC 1109	Computer Literacy	4	30	30	30	60	Old
	Total CU					**17**		**Toal CU**	**19**					
Year 1 Semester 2							**Year 1 Semester 2**							
BSE1206	Software Development Principles	45	-	30	60	4	BSE1206	Software Development Principles	3	45	-	-	45	Old
MTH2203	Numerical Analysis I	45	-	30	45	3	MTH2203	Numerical Analysis I	3	30	-	30	45	Old
BSE1207	Introduction to Internet Programming	30	30	30	60	4	BSE1209	Object Oriented Programming I	4	30	30	30	60	New
BIS1204	Data and Information Management I	45	-	30	60	4	IST1203	Data and Information Management I	4	30	60	-	60	Old
BIS1206	Systems Analysis and Design	30	60	-	60	4	BSE1208	Introduction to Web Development	4	30	30	30	60	New
	Total CU					**19**		**Total CU**	**18**					
Year 1 Recess Term							**Year 1 Recess Term**							
BSE1302	Professional Software Engineering Mini Practical Project I	-	60	60	60	4	BSE 1302	Software Engineering Practical Skills Project I	5	-	150	-	75	Old
Year 2 Sem 1								**Year 2 Sem 1**						
CSC2100	Data Structures and Algorithms	45	-	30	60	4	CSC2114	Artificial Intelligence	4	30	30	30	60	Modified
MTH3105	Discrete Mathematics	30	-	30	645	3	CSC2100	Data Structures and Algorithms	4	30	30	30	60	Old
CSC2114	Artificial Intelligence	30	-	30	645	3	BSE2106	Computer Networks	4	30	30	30	60	Old
BSE2106	Computer Networks	45	30	-	60	4	BSE2105	Formal Methods	3	45	-	-	45	Old
	Electives						BSE2107	Object Oriented Programming II	4	30	30	30	60	New
CSC1104	Computer Organization and Architecture	60	-	-	60	4								
BSE2105	Formal Methods	30	30	-	45	3								
	Total CU					**18**		**Total CU**	**19**					
Year 2 Sem 2								**Year 2 Sem 2**						

CSC2200	Operating Systems	45	-	30	60	4		CSC2200	Operating Systems	4	30	30	30	60	Modified
BSE2205	Network Application Development	45	30	-	60	4		BSE2207	Emerging Web Development Technologies	4	30	30	30	60	New
BSE2206	Data Communications	45	30	-	60	4		BSE2206	Data Communication	3	45	-	-	45	Modified
MTH1203	Calculus 1	45	-	30	60	4		BSE2209	Mobile Programming Project	4	15	60	30	60	New
CSC1214	Object Oriented Programming	30	60	-	60	4		BSE2208	Requirements Engineering	4	45	30	30	60	Old
	Total CU					**20**			**Total CU**	**19**					
Year 2 Recess Term									**Year 2 Recess Term**						
BSE2302	Professional Software Engineering Mini Practical Project II	-	60	60	60	4		BSE2302	Software Engineering Practical Skills Project II	5	-	150	-	75	Old
Year 3 Sem 1									**Year 3 Sem 1**						
BSE3110	Object-oriented Analysis and Design	45	30	-	60	4		BSE3110	Object Oriented Analysis and Design	4	30	30	30	60	Old
BSE3111	Requirements Engineering	45	-	30	60	4		BSE3111	Embedded Systems I	4	30	30	30	60	New
BSE3104	Software Metrics	30	-	30	45	3		BSE3104	Software Metrics	3	45	-	-	45	Old
CSC3119	User Interface Design	45	30	-	60	4		CSC3110	User Interface Design	4	30	30	30	60	Modified
	Electives								**Electives**						
BSE3106	Mobile Networks and Computing	45	30	-	60	4		BSE3106	Mobile Networks and Computing	4	45	30	-	60	old
BSE3109	Introduction to Embedded Systems	45	-	30	60	4		BSE3105	Software Evolution	4	45	-	30	60	Old
BSE3108	Communications Systems Design	45	30	-	60	4									
BSE3105	Software Evolution	45	-	30	60	4									
	Total CU					**19**			**Total CU**	**18**					
Year 3 Sem 2									**Year 3 Sem 2**						
BSE3201	Software Architecture	45	-	30	60	4		BSE3210	Software Architecture and Patterns	3	45	-	-	45	New
BSE3208	Distributed Systems Development	45	30	-	60	4		BSE3211	Software Testing and Verification	3	45	-	-	45	New
BSE3209	Advanced Object-Oriented Programming	30	60	-	45	4		CSC2206	Machine Learning	4	45	-	30	60	Old
BIT2207	Research Methodology	30	-	30	45	3		IST2203	Research Methodology	4	30	30	30	60	Old
	Electives								Electives						
BIT2208	Systems Administration	45	30	-	60	4		BSE3214	Cloud Computing and Big Data	4	30	30	30	60	New

CSC2209	Systems Programming	45	-	30	60	4		BSE3213	Embedded Systems II	4	30	30	30	60	New
	Total CU					**19**			**Total CU**	**18**					
Year 3 Recess Term									**Year 3 Recess Term**						
BSE3302	Field Attachment	-	-	120	60	4		BSE 3302	Field Attachment	5	-	150	-	75	Old
Year 4 Sem 1									**Year 4 Sem 1**						
BSE4100	Software Engineering Project I	-	-	150	75	5		BSE4100	Software Engineering Project I	5	-	150	-	75	Old
BSE4101	Software Reliability and Testing	45	-	30	60	4		BSE4102	ICT Innovation and Entrepreneurship	3	45	-	-	45	New
BIS3106	Business Process Management	45	30	-	60	4		BSE4104	Emerging Trends in Software Engineering	3	45	-	-	45	Modified
BAM2102	Entrepreneurship Principles	45	30	-	60	4		BSE4105	Software Integration and Deployment	4	30	60	-	60	New
	Total CU					**17**			**Total CU**	**15**					
Year 4 Sem 2															
BSE4200	Software Engineering Project II	-	-	150	75	5		BSE4200	Software Engineering Project II	5	-	150	-	75	Old
BSE4202	Software Security	45	30	-	60	4		BSE4202	Software Security	4	30	60	-	60	Modified
BSE4201	Software Design Patterns	45	30	-	60	4		BSE4203	Software Engineering Standards and Ethics	3	45	-	-	45	New
BIT2209	IT Law and Ethics	45	-	30	45	3		BSE4204	Software Quality Management	3	45	-	-	45	New
	Total CU					**16**			**Total CU**	**15**					

2 Programme Description

2.1 Admission Requirements

To qualify for admission to the B.Sc. in Software Engineering (BSSE), a candidate must fulfill the general Makerere University entry requirements for Bachelor's degree, and in addition the candidate wil be admited **to first year** through three (3) avenues, namely; Direct entry, Diploma entry and degree entry schemes.

2.1.1 Direct Entry

A candidate must satisfy the general minimum entry requirements of Makerere University. In addition, the following regulations shall hold: Candidates seeking admission through this avenue must have obtained:

1. The Uganda Certificate of Education (UCE) or its equivalent, with atleast 5 passes and a credit in English and Mathematics.
2. At least two principal passes attained from the same sitting in Uganda Advanced Certificate of Education (UACE) in Mathematics and any of Physics, Economics, Chemistry, Biology, Geography.
3. For purposes of computing weighted points, the advanced level subjects shall be grouped as follows.

Group	Weight	Subject
Essential	3	Mathematics and one best done of the following subjects: Physics, Chemistry, Economics, Geography, Biology
Relevant	2	The third best done of the following subjects: Physics, Chemistry, Economics, Geography, Biology
Desirable	1	General Paper, Sub-Mathematics, ICT
Others	0.5	All others.

2.1.2 Diploma Holders

For a candidate to be admitted via the diploma 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 5 passes got at the same sitting from a recognized academic institution, with credits in Mathematics, information technology, statistics, computing, programming or engineering related courses.
2. At least a Second class (lower division) diploma in Software Engineering, Computer Science, Mathematics, Engineering, Information Technology, and Statistics or any other diploma with Software/applications development, Mathematics, Computer Science, or Information Technology as one of the subjects.
3. The diploma must be attained from an Institution recognized by the National Council for Higher Education in Uganda.

2.1.3 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, Economics, 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.

2.2 Target Group

The programme targets three categories of people, namely A level certificate holders and Diploma holders and Degree holders in relevant programmes.

The programme targets to enroll 100 students on day and 100 students on evening and hence a total of 200 students.

2.3 Nature of the Programme

This programme cater is a day programme for all government sponsored studdents and caters for both day and evening private sponsored students.

2.4 Programme Duration

The duration for the BSc. in Software Engineering degree programme is four (4) academic years comprising 8 semesters, two practical skiling recess terms in first and second year, and a field attachment session in third year.

3 Resources and Infrastructure

3.1 Financial Resources

The curriculum for the BSSE is mainly funded and sustained by tutition fees as distributed in budget appendix B. Ugandan (or East African[2]) students are required to pay tuition fees totalling to three Million twenty- four thousand Ugandan shillings (3,024,000 UGX) per year. Foreign students are required to pay tuition fees totaling to $3700 per year. Other financial resources include grants from research projects and short courses at the CiPSD(Center for *i*nnovations and Professional Skills Development) as well as funded industrial research done by the research groups. Recess and field attachment will be paid separate from the academic semester tuition fees.

3.2 Administration and Technical Support

The department has a head assisted by several coordinators of departmental activities. Further, the department has an administrative assistant to help on different queries that may be raised by students.

To further support community and private partnership, the College has a Communications Office that links students and staff to the public. In addition, the College has a Workforce development Office to help and link students to potential employs.

[2] Compliance with Uganda government regulations

In relation to practicals and labs, the College has a dedicated ICT services unit to ensure that all ICT facilities are up and running. For student who have their own laptops, there is a wireless access area which is accessible by students anytime. Moreover, each of the Labs has a dedicated Lab assistant and Computer engineer to help students during their free time outside lecture hours.

3.3 Human Resource (Academic Staff)

The Department of Networks has sufficient academic staff to run this programme. Refer to Appendix A for a staff list. Secondly, some cross-cutting courses will be supported by staff from other departments in the School of Computing and Informatics Technology. Appendix A is a list of academic staff.

3.4 Physical Facilities

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 is mainly administrative, accommodating offices for the administration and teaching staff. The ground floor has a conferencing facility co-shared with AMITY University; Reception at first floor while second floor houses the Information Technology department, Finance and procurement offices, a teaching lab and a lecture room. Third floor houses the networks department, Information systems department, a seminar room for meetings/teaching purposes and Computer sciences department offices for staff while the fourth floor has well furnished seminar room and conference hall, offices including: Dean school of computing and IT, the college principal and deputy principal's offices as well as their administrative assistants, all heads of departments and the kitchen facility, Fifth floor houses the store (logistics and stationary), office of the human resource and also office of the administrative assistants to the dean and heads of department. The sixth floor has a testing centre.

Block B has lecture theatres together the rest of the general and specialized laboratories i.e. Software Development Centre, Multimedia lab, Mobile Applications, students' computer labs, among others. The college accounts and registrar's offices are on level 3 while level 5 houses the Center for Innovations and Professional Skills Development (CiPSD). The main college canteen is on ground floor of the block. The two buildings sufficiently cater for all the lecture and laboratory space requirements at CoCIS. Specifically, CoCIS has thirteen lecture theatres each of 200 square meters (300 seat capacity) of circulation space where students are able to access other services such wireless Internet services.

3.5 Computing Equipment 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.

3.6 Use of ICT in availing lecture materials

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

3.7 Library

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 ac- cess 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 at http://196.43.133.123:8080.

Below is a list of all electronic databases that Makerere subscribes to;

a) Institute of Electrical and Electronic Engineers (IEEE)
b) Emerald Insight
c) Springer Verlag
d) Research4life (ARDI & HINARI)
e) Sage Publications
f) E-library (eBook database)
g) Science Direct

4 Programme Regulations

Here we give regulations specific to the programme. Additional normal regulations that relate to illness, absence from the program conduct of examinations can be found in the undergraduate hand- book available at Academic registrar's office and School of Computing and Informatics Technology.

4.1 Grading of Courses

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

MARKS %	LETTER GRADE	GRADE POINT	INTERPRETATION
90 – 100	A+	5.0	Exceptional
80 – 89	A	5.0	Excellent
75 – 79	B+	4.5	Very Good
70 – 74	B	4.0	Good
65 – 69	C+	3.5	Fairly good
60 – 64	C	3.0	Fair
55 – 59	D+	2.5	Pass
50 – 54	D	2.0	Marginal Pass
45 – 49	E	1.5	Fail
40 – 44	E-	1.0	
Below 40	F	0	

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

W	-	Withdraw from Course;
I	-	Incomplete;
AU	-	Audited Course Only;
P	-	Pass;
F	-	Failure.

4.2 Minimum Pass Mark

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

4.3 Calculation of Cumulative Grade Point Average (CGPA)

The CGPA shall be calculated as follows: -

$$\text{CGPA} = \frac{\sum_{i=1}^{n}(GP_i \times CU_i)}{\sum_{i=1}^{n} CU_i},$$

where *GPi* is the Grade Point score of a particular course *i*; *CUi* is the number of Credit Units of course *i*; and *n* is the number of courses so far done.

4.4 Progression

Progression through the programme shall be assessed in three ways, namely; normal, probationary or discontinuation as per the standard Makerere University Senate guidelines.

4.4.1 Normal Progress

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

4.4.2 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 waved when these conditions cease to hold.

4.4.3 Discontinuation

When a student accumulates three consecutive probations based on the CGPA or the same core course unit(s), s/he shall be discontinued. A student who has failed to obtain at least the pass mark of 50% or Grade Point of 2.0 after the third attempt in the same course unit(s) s/he had retaken shall be discontinued from his/her studies at the University. A student who has overstayed on an academic program by more than two (2) years shall be discontinued from his/her studies at the University.

4.5 Weighting System

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

4.6 Minimum Graduation Load

To qualify for the award of the degree of BSSE a full-time, a candidate is required to obtain a minimum of 156 CUs from all core courses, three core recess terms and the elective courses as indicated in the program structure.

4.7 Knowledge Areas Covered in the Programme

The following knowledge areas are covered in the revised BSSE Curriculum;

KA1 - Foundation in Software Engieering

KA2 - Software requirements

KA3 - Software design

KA4 - Software construction

KA5 - Software testing and verification

KA6 - Application Conectivity

KA7 - Software configuration management

KA8 - Software engineering process management

KA9 - Lifelong skills

KA10 - Standards and ethics

Year / sem	KA1	KA2	KA3	KA4	KA5	KA6	KA7	KA8	KA9	KA10
Year1 sem 1	IST1101 BSE1107 CSC1109		BSE1108						UNV1101 BSE1106	
Year1 sem 2			IST1203	BSE1206 BSE1209 BSE1208					MTH2203	
Recess		BSE1302		BSE 1302						
Year 2 Sem 1							CSC2100			
Year 2 Sem 2		BSE2208				BSE2206				
Recess										
Year 3 Sem 1				BSE3111 CSC3110				BSE3104 BSE3105		
Year3 sem 2			BSE3214	BSE3213	BSE3211					
Field Attach								BSE3302	BSE3302	
Year 4 Sem 1				BSE4100					BSE4102	BSE4104
Year 4 Sem 2					BSE4204					BSE4200 BSE4202 BSE4203

5 Programme Structure of the B.Sc. Software Engineering

The revised BSSE degree programme is a day and evening programme covered in a period of four academic years, each year having two semesters of 17 weeks and a recess term of 10 weeks(in the first three years). The proposed programme requires a minimum graduation load of 157 credit units and a semester shall have a minimum load of 15 credit units.

The details of the programme structure are shown below, where LH, PH, CH and CU stand for Lecture Hours, Practical Hours, Contact Hours and Credit Units respectively. The remarks provided in the program structure below mean the following;

- **Modified** means that the old curriculum course code and name have been retained but the course content has been modified to meet the objectives of the revised curriculum.
- **Old** means that both course code, course name and content of the old curriculum have been retained.
- **New** means that the course is completely new in this revised curricullum.

The mother department where each course unit is hosted/administered is also indicated in the programme structure below where NW means Networks department, IS means Information Systems department, CS means Computer Science department and IT means Information Technology department.

5.1 BSSE Programme Structure

CODE	COURSE NAME	CU	LH	PH	TH	CH	Type	Remark	Origin
Year 1 - Semester I (5 Core Courses, No Electives) - Total Credit Units=19									
BSE1106	Problem Solving and Programming Concepts	4	30	30	30	60	Core	Modified	NW
UNV1101	Communication Skills	4	30	60		60	Core	Old	Langu
BSE1107	Mathematics for Software Engineers	3	30	-	30	45	Core	New	NW
BSE1108	Technical Analysis and Design	4	30	30	30	60	Core	New	NW
IST1101	Foundations of Information Systems & Technology	4	45	30	-	60	Core	Old	IS
	Audit Course								
CSC 1109	Computer Literacy	4	30	30	30	60	AU	Old	CS
Year 1 - Semester II (5 Core Courses, No Electives) - Total Credit Units=18									
BSE1206	Software Development Principles	3	45	-	-	45	Core	Old	NW
MTH2203	Numerical Analysis I	3	30	-	30	45	Core	Old	Math
BSE1209	Object Oriented Programming I	4	30	30	30	60	Core	New	NW
IST1203	Data and Information Management I	4	30	60	-	60	Core	Old	IS
BSE1208	Introduction to Web Development	4	30	30	30	60	Core	New	NW
Year I Recess (1 Core Course, No Electives) – Total Credit Units = 5									
BSE 1302	Software Engineering Practical Skills Project I	5	-	150	-	75	Core	Old	NW
Year 2 - Semester I (5 Core Courses, No Electives) - Total Credit Units=19									
CSC2114	Artificial Intelligence	4	30	30	30	60	Core	Modifie	CS
CSC2100	Data Structures and Algorithms	4	30	30	30	60	Core	Old	CS
BSE2106	Computer Networks	4	30	30	30	60	Core	Old	NW

BSE2105	Formal Methods	3	45	-	-	45	Core	Old	NW
BSE2107	Object Oriented Programming II	4	30	30	30	60	Core	New	NW
Year 2 - Semester II (5 Core Courses, No Elective) - Total Credit Units=19									
CSC2200	Operating Systems	4	30	30	30	60	Core	Modified	CS
BSE2207	Emerging Web Development Technologies	4	30	30	30	60	Core	New	NW
BSE2208	Requirements Engineering	3	45	-	-	45	Core	Modified	NW
BSE2209	Mobile Programming Project	4	15	60	30	60	Core	New	NW
BSE2206	Data Communication	4	45	30	30	60	Core	Old	NW
Year II Recess(1 Core Course, No Electives) – Total Credit Units = 5									
BSE2302	Software Engineering Practical Skills Project II	5	-	150	-	75	Core	Old	NW
Year 3 - Semester I (5 Core Courses) - Total Credit Units=18									
BSE3110	Object Oriented Analysis and Design	4	30	30	30	60	Core	Old	NW
BSE3111	Embedded Systems I	4	30	30	30	60	Core	New	NW
BSE3104	Software Metrics	3	45	-	-	45	Core	Old	NW
CSC3110	User Interface Design	4	30	30	30	60	Core	Modified	CS
	Electives								
BSE3106	Mobile Networks and Computing	4	45	30	-	45	Elective	old	NW
BSE3105	Software Evolution	4	45	-	30	60	Elective	Old	NW
Year 3 - Semester II (5 Core Courses, Choose ONLY 1 Elective) - Total Credit Units=18									
BSE3210	Software Architecture and Patterns	3	45	-	-	45	Core	New	NW
BSE3211	Software Testing and Verification	3	45	-	-	45	Core	New	NW
IST2203	Research Methodology	4	45	-	30	60	Core	Old	IT
CSC2206	Machine Learning	4	30	30	30	60	Core	Old	CS
	Electives								
BSE3214	Cloud Computing and Big Data	4	30	30	30	60	Elective	New	NW
BSE3213	Embedded Systems II	4	30	30	30	60	Elective	New	NW
Year III Recess(1 Core Course, No Electives) – Total Credit Units = 5									
BSE 3302	Field Attachment	5	-	150	-	75	Core	Old	NW
Year 4 - Semester I (4 Core Courses, No Electives)- Total Credit Units=15									
BSE4100	Software Engineering Project I	5	-	150	-	75	Core	Old	NW
BSE4102	ICT Innovation and Entrepreneurship	3	45	-	-	45	Core	New	NW
BSE4104	Emerging Trends in Software Engineering	3	45	-	-	45	Core	Modified	NW
BSE4105	Software Integration and Deployment	4	30	60	-	60	Core	New	NW
Year 4 - Semester II (4 Core Courses, No Electives)- Total Credit Units=15									
BSE4200	Software Engineering Project II	5	-	150	-	75	Core	Old	NW
BSE4202	Software Security	4	30	60	-	60	Core	Modified	NW
BSE4203	Software Engineering Standards and Ethics	3	45	-	-	45	Core	New	NW
BSE4204	Software Quality Management	3	45	-	-	45	Core	New	NW

6 Detailed Curriculum

1.1 Year I Semester I

1.1.1 BSE1106 Problem Solving and Programming Concepts

Course Level : Year I Semester I
Course Credit : 4
Contact Hours : 60

i. **Course Description.**

The course trains students from different backgrounds how to design programs. A Student is taught how to articulate thoughts about a program. The course gives design guidelines that lead students from a problem statement to a computational solution in step-by-step fashion with well-defined intermediate products.

ii. **Course Objectives.** The course objectives include;

- Equip students with skills of critical reading, analytical thinking, creative synthesis, attention to detail and writing basic structured programs.
- Impart knowledge of C programming language to implement programming concepts.

iii. **Learning outcomes.**

After successfully completing this course student will be able to:

- Read, analyze, organize, experiment, and think in a systematic manner.
- Write simple programs from algorithm,
- Demomonstrate learned tools for designing algorithms from human understanding to computer understanding.
- Write simple programs in C.

iv. **Prerequisites.** None

v. **Indicative Content**

- Introduction to problem solving techniques, problem solving with flow charts, algorithms and their interpretation. (10 hours)
- Programming language classifications (20 hours)
  - o By abstraction level (Low level, high level, very high level),
  - o By domain (business languages, scientific languages, AI languages, systems languages, scripting languages, XML-based languages),
  - o By generality (general purpose vs. special purpose), By implementation methods (Interpreted vs. compiled),
  - o By paradigm (imperative, object-oriented, logic-based, functional).
- Elements of Programming Languages using C (15 hours)
  - o Syntax, Semantics, Data
  - o Programming environments (editors, compilers, linker, loader, debugger, interpreters etc)
- Program design guidelines that show the reader how to analyze a problem statement; (6 hours)
- Formulation of concise goals; (9 hours)
  - o How to make up examples; how to develop an outline of the solution, based on the analysis;
  - o How to finish the program; and how to test. The logical and physical structure of programs and data.

vi. **Mode of delivery**

Teaching and learning will be in form of classroom lectures, demonstrations

students practical projects, class discussions and quiz
To be taught with ODeL provision

vii. **Assessment Method**

- Course work (Tests, group assignments (take home assignments, case studies, individual / group projects)) 30%
- Final written exam: 70%

viii. **References**

- Matthias Felleisen, Robert Bruce Findler, Matthew Flatt , Shriram Krishnamurthi, 2018 How to Design Programs: An Introduction to Programming and Computing. 2nd Ed., The MIT Press, ISBN-13 : 978-0262534802
- Maureen Sprankle and Jim Hubbard, 2012, Problem solving and Programming Concepts, 9th Ed, Pearson Education, New Delhi, , ISBN-13: 9780132492645
- Anton Spraul, 2012 Think Like a Programmer: An Introduction to Creative Problem Solving.

1.1.2 UNV1101 Communication Skills

Course Level : Year I Semester I
Course Credit : 4
Contact Hours : 60

i. 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.

ii. 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.

iii. 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.

iv. **Prerequisites**: None

v. Indicative Content

- Topic 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.

- Topic 2: Reading Skills (20 Hours)

The use of skimming; scanning inference and prediction in reading; Intensive and critical reading; Acquisition of speci_c reading skills; Interpretation of non-linear texts; Locating information and comprehension.

- Topic 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 clari_cation etc.; Expression of politeness; Public speaking; Proper listening skills.

vi. Mode of delivery

This course will involve 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.

vii. Assessment

Course work assessment will contribute (30%) and examination will contribute (70%).

viii. References

- James W. Williams 2020, Communication Skills Training: How to Talk to Anyone, Connect Effortlessly, Develop Charisma, and Become a People Person, ISBN13 : 979-8646165757
- Courtland L. Bove and John V. Thill, 2005, Business Communication Today. 8th edition. Upper Saddle River, NJ: Prentice Hall International Inc., ISBN-13 : 978-0131478459
- Ian Tuhovsky and Wendell Wadsworth 2008, Communication Skills: A Practical Guide to Improving Your Social Intelligence, Presentation, Persuasion and Public Speaking (Positive Psychology Coaching Series Book) V9, 1st ed, ISBN-13: 978-1515031918
- Grant, A.E. Meadows, Focal J.H., 2014, Communication Technology Update and Fundamentals ,Focal Press.

1.1.3 BSE1107 Mathematics for Softeware Engineers

Course Level : Year I Semester I
Course Credit : 3
Contact Hours : 45

i. Course Description.

A good grounding in mathematics provides a good foundation for a software engineer. This course is meant to equip the student with good computational skills in preparation for computational courses under this program.

ii. Course Objectives.

At the end of the course:

Students will understand the mathematics required for software engineering. They will be able to reason about the mathematical objects that appear in a specification. They will also gain an appreciation for varying degrees of rigor and formality, choosing an appropriate level of detail or elaboration.

iii. Learning outcomes.

At the end of the course, it is expected that the student is eable to;

- Construct syntactic and semantic proofs in propositional and predicate logic.
- Express English language and mathematical expressions in logic.
- Determine semantic equivalences, satisfiability and validity.
- Verify the correctness of computer programs.
- Apply model checking for verification.

iv. **Prerequisites.** Although no previous experience is required, a positive attitude towards formal, mathematical notation would be an advantage.

v. **Indicative Content**

- **Propositional logic:** propositions; logical connectives; deductive reasoning; hypothetical reasoning (5 hours)
- **Predicate logic:** quantification; substitution; scope of variables; generalization; specialization (5 hours)
- **Equality:** definite description; the one-point rule; uniqueness and quantity (5 hours)
- **Sets:** membership; extension; comprehension; power sets; Cartesian products; types (5 hours)
- **Definitions:** basic types; declarations; abbreviations; axioms; generics; consistency (5 hours)
- **Relations:** domains and ranges; projections; inverses; compositions; iteration; closure (5 hours)
- **Functions:** partial functions; injections; surjections; lambda notation; overriding; enumerations (5 hours)
- **Sequences:** order and multiplicity; sequence operators; sequences as functions; structural induction; bags (5 hours)
- **Free types:** constants and constructor functions; embedding; closure; induction principles (5 hours)

vi. **Mode of delivery**

Lectures, class discussions and quiz

vii. **Assessment Method**

- Course work (Tests, group assignments (take home assignments, case studies, individual / group projects)) 30%
- Final written exam: 70%

viii. **References**

- Paul Orland, 2020, Math for Programmers 3D graphics, machine learning, and simulations with Python, ISBN 9781617295355
- Jeremy Kun, 2018, A Programmer's Introduction to Mathematics 1st Edition, CreateSpace Independent Publishing Platform; ISBN-10 : 1727125452
- Martin J. Loomes, Jim Woodcock, 1988, Software Engineering Mathematics 1st Edition, CRC Press; 1st edition, ISBN-10 : 0273026739
- Jim Woodcock, Martin J Loomes, 1990, Software engineering mathematics, Addison-Wesley Longman Publishing Co., Inc.75 Arlington Street, Suite 300 Boston, MA, United States, ISBN:978-0-201-50424-8
- Discrete Mathematics: An Open Introduction by Oscar Levin ,2018, BN-13: 978-1792901690, 3rd Edition.

1.1.4 BSE1108 Technical Analysis and Design

Course Level : Year I Semester I
Course Credit : 4
Contact Hours : 60

i. Course Description

This course will create a strong base in the principles and practice of software systems design. A high level-modelling tool like UML will be used in this course. Students will cover both theoretical principles and hands on practical skills group assignments. The purpose of this course is to familiarize students with concepts, methods, and tools for object oriented analysis and design of software systems with practical applications in large product development projects.

ii. Course Objectives

The objectives of this course include;

- To impart concepts, methods, and tools for object oriented analysis and design of software systems
- To teach students the methods applied in large product development projects.

iii. Learning outcomes

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

- explain how to represent a software system using UML models,
- explain guidelines and heuristics for performing a domain analysis,
- analyze and design software systems using object oriented techniques,
- create an UML model that is an abstract representation of the source code,
- use tools for domain and requirements analysis, modeling, program visualization, and object oriented program design.
- analyze how software design principles and patterns impact software quality,
- reflect on - and resolve inconsistencies between various models used as part of a single system's design.

iv. Prerequisites. None

v. Indicative Content

- Introduction to common design principles (6 hours)
- Patterns that support the development of maintainable, reusable and extensible software. (6 hours)
- Identifying the Major Trend in technical analysis and design (9 hours)
- Introduction to UML. (3 hours)
- Expression of Analysis and design models using UML models. (9 hours)
- Models like use case diagrams, class diagrams, sequence diagrams, and state diagrams.
- Introduction to techniques and guidelines for analysis of software domain and requirements (6 hours)
- Derived and quantitative analysis (6 hours)
- Indicators of good analysis and design (9 hours)
- Option strategies and combination of options (6 hours)

vi. Mode of delivery

Relevant textbooks, UML Design software like Microsoft Visio, power point slides, a projector, laptop and necessary teaching aids. The mode of delivery will be lectures, group work for mini project assignments, class exercises and test, which will be graded to contribute to the coursework scores.

vii. Assessment

- Mini-project assignment and one test (30%)
- Final written examination (70%)

viii. References

- Joseph Valacich, Joey George, 2020, Modern Systems Analysis and Design 9th Edition, Pearson, ISBN-13 : 978-0135172759
- Larry Christensen, R. Johnson , Lisa Turner, 2013 Research Methods, Design, and Analysis 12th Edition, Pearson, ISBN-13 : 978-0205961252
- Larman, Craig, 2012, Applying UML and patterns: an introduction to object oriented analysis and design and interative development. Pearson Education India.
- Greg Harmon 2014, Trading Options: Using Technical Analysis to Design Winning Trades, + Website, Wiley Trading, ISBN: 978-1-118-67913-5

1.1.5 IST1101 Foundations of Information Systems and Technology

Course Level : Year I Semester I
Course Credit : 4
Contact Hours : 60

i. Course Description

This course is designed to provide BIST students with an overview of the IST discipline so that they are grounded in fundamental aspects of Information Systems and Technology. To achieve that, the course is divided into two parts. The first part exposes students to contemporary information systems and demonstrates how these systems are used throughout global organizations. The focus of the first part is on key components of information systems – people, software, hardware, data, and communication technologies and how these can be integrated and managed to create competitive advantage. Through the knowledge of how Information Systems provide a competitive advantage, students will gain an understanding of how information is used organizations and how IT enables improvements in quality, speed, and agility. The second part of the course exposes students to IT infrastructure and covers topics related to both computer and systems architecture and communication networks, with an overall focus on the services and capabilities that IT infrastructure solutions enable in an organizational context.

ii. Course Objectives

The objectvies of the course include;

1. To expose students to the technology, people, and organizational components of information systems, and how and why information systems are used today.
2. To teach students to the value of information systems investments, and how businesses are using them for competitive advantage.
3. To introduce students to principles underlying layered systems architectures and their application to both computer and network systems and how they enable new forms of communication, collaboration, and partnering.
4. To expose students to how various types of information systems provide the information needed to gain business intelligence to support decision making for different levels and functions of the organization.
5. To elaborate how enterprise systems foster strong relationships with customers and suppliers and how these systems are widely used to enforce organizational structures and processes.

6. To expose students to the components and structure of a large-scale organizational IT infrastructure solution at a level that allows them to use it effectively; the role of IT control and service management frameworks in managing large scale organizational IT infrastructure and solution and opportunities that virtual computing service models, such as cloud computing create for organizations.

iii. Learning outcomes

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

- Explain the technology, people and organizational components of information systems, and how and why information systems are used today.
- Describe the value of information systems investments and explain correctly how businesses are using information systems for competitive advantage.
- Demonstrates how information systems are enabling new forms of communication, collaboration, and partnering.
- Describe how various types of information systems provide the information needed to gain business intelligence to support decision making for different levels and functions of the organization.
- Explain how enterprise systems foster strong relationships with customers and suppliers and how these systems are widely used to enforce organizational structures and processes.
- Identify opportunities that virtual computing service provision models, such as cloud computing create for organizations and analyze the security and business continuity implications of IT infrastructure design solutions.

iv. **Prerequisites**. None

v. Indicative Content

Part I

1.	Characteristics of the digital world	2 Hours
2.	Information Systems components	3 Hours
3.	Information Systems in Organizations	3 Hours
4.	Globalization	3 Hours
5.	Information Systems Infrastructure	3 Hours
6.	Valuing information systems	2 Hours
7.	Information systems for supporting decision making for different levels and functions of an organization	4 Hours
8.	Enterprise-wide information systems	4 Hours

Part 2

1.	Core computing system architecture concepts	3 Hours
2.	Core computing system organizing structures	3 Hours
3.	Core technical components of computer based systems	3 Hours
4.	Role of IT infrastructure in a modern organization	3 Hours
5.	Role of IT control and service management frameworks	3 Hours
6.	Ensuring business continuity	2 Hours
7.	Grid computing, cloud computing, computing as services	4 Hours
8.	Purchasing of IT infrastructure technologies and services	3 Hours

vi. Mode of delivery

Using text books, conference and journal publications, and online resources. Model of delivery will include Lectures, Online learning management systems, Class discussions and presentations, Problem-based/case studies and Project-based assignments that can be done in groups.

To be taught with ODeL provision

vii. Assessment

- Course work 30%
- Final written exam 70%

viii. References

1. Stair, R. & Reynolds, G. (2013). *Foundations of Information Systems*, 7th Edition. Publisher: Cengage South-Western.
2. Valacich J. and Schneider, C. (2012). Information Systems Today, 5th Edition. Publisher: Prentice Hall.
3. Laan, S. (2011). IT infrastructure architecture – Infrastructure building blocks and concepts. 1st Edition. Publisher: Lulu Press Inc.
4. Choubey, M. K. (2012). IT Infrastructure and Management. 1st Impression. Publisher: Dorling Kindersley (India).

1.1.6 CSC1109 Computer Literacy

Course Level	: Year I Semester I
Course Credit	: 4
Contact Hours	: 60

i. 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.

ii. Course Objectives

The objectives of this course are, to:

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

iii. Learning Outcomes

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

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

iv. Detailed Course Content

1. General computer organization	(12 Hours)
2. Historical perspectives of computing	(8 Hours)
3. Common Microsoft office packages	(12 Hours)

4. Office packages in other operating systems	(8 Hours)
5. Text editors	(4 Hours)
6. Common Command Line commands (Linux, Dos)	(8 Hours)
7. Using the web	(4 Hours)
8. Understanding the Internet	(4 Hours)

v. Mode of delivery

- Lectures (with ODeL provision)
- 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

vi. Mode of Assessment

The assessment will be in form of:

- Course work (Tests, group assignments (take home assignments, case studies, individual / group projects)) 30%
- Final written exam: 70%

vii. Reading List

- Gerardus Blokdyk, 2020, Computer Literacy A Complete Guide, 5STARCooks, ASIN : B084FP65VN
- Preston, J. Preston, S. & Ferrett, R. Computer Literacy: Computing Fundamentals. Prentice Hall Press., 2012.
- Snyder, L. 2008. Fluency with information technology: Skills, Concepts, and Capabilities. John Wiley & Sons, Inc., 2008

1.2 Year I Semester II

1.2.1 BSE1206 Software Development Principles

Course Level : Year I Semester II
Course Credit : 3
Contact Hours : 45

i. Course Description

This course introduces students to the basic software Development Principles & Processes highlighting the development of software systems with a focus on leading software development and management processes and practices. This is design to orient a student a Systems Engineering perspective, that is, an interdisciplinary, collaborative approach to the engineering of system solutions which aims to capture stakeholder needs and objectives and to transform these into a holistic, life-cycle balanced system solution which both satisfies the minimum requirements of the stakeholders, and optimises overall solution effectiveness according to the values of the stakeholders.

ii. Course Objectives

The objectives of this course are, to;

- Enable the student to specify, plan, develop, deliver, maintain and operate a software intensive system.
- Demonstrate to the students the practical methods for improving software performance in an organization.
- Practically guide the student on the application of exitsting software development standards

iii. Learning outcomes

At the conclusion of this course the student will:

- be better prepared to specify, plan, develop, deliver, maintain and operate a software intensive system. Given the constant evolution of software development methodologies, delegates who are already experienced software professionals will further develop their skills.
- become an advocate in your organization of practical methods to improve software project performance.
- identify causes of software development problems and drive project performance inprovements.
- value the substantial body of public domain knowledge that defines world's leading practice in software engineering. For example, ISO/IEC/IEEE 12207, other IEEE and ISO standards, the Software Engineering Institute (SEI) Capability Maturity Models and the Guide to the Software Engineering Body of Knowledge (SWEBOK).

iv. Mode of delivery

Teaching will be in terms of lectures, problem–based/case studies, group work, and presentations.

To be taught with ODeL provision

v. Prerequisite: None

vi. Indicative content

1.	Introduction to software development.	(3 hours)
2.	Why is Systematic Software Development necessary?	(3 hours)
3.	Phases of System Development Lifecycle	(3 hours)
4.	System development approaches	(3 hours)
5.	System development methodologies or models	(3 hours)
6.	Software Systems Engineering Process Frameworks	(3 hours)
7.	Agile Methods and Techniques	(3 hours)
8.	Software Design & construction	(6 hours)
9.	System Integration	(4 hours)
10.	Quality Management	(4 hours)
11.	Technical reviews	(3 hours)
12.	Other verification and validation approaches and methods	(3 hours)
13.	Project Management Frameworks and activities	(4 hours)

vii. Assessment

- Course work (Tests, group assignments (take home assignments, case studies, individual / group projects)) 30%
- Final written exam: 70%

viii. References

- Roger Pressman and Bruce Maxim, 2020, Software Engineering: A Practitioner's Approach 9th Edition, ISBN10: 1259872971, ISBN13: 9781259872976
- Brian Kernighan and Dennis Ritchie, C Programming, 3rd Edition
- Davis, A. M. 1995, 201 Principles of Software Development. (New York, NY: McGraw-Hill, Inc.)
- Brett McLaughlin, Gary Pollice, David West, 2007 *Head First Object-Oriented Analysis & Design*. (Sebastopol, CA: O'Reilly Media, Inc.)

1.2.2 MTH2203 Numerical Analysis I

Course Level : Year I Semester II

Course Credit : 3
Contact Hours : 45

i. Course Description

Numerical Analysis is the study of algorithms that use numerical approximation (as opposed to general symbolic manipulations) for solving problems of Mathematical Analysis. It therefore, an approach to solving complex mathematical problems using simple approximating operations and carrying out an analysis on the resulting errors. Numerical Analysis plays an indispensable role in solving real life mathematical, physical and engineering problems. Numerical computations have been in use for centuries even before digital computers appeared on the scene.

Great mathematicians like Gauss, Newton, Lagrange, Fourier and many others developed numerical techniques. In this course, the following areas will be covered: finite differences, interpolation, differentiation, integration, solution of non-linear equations and solution of systems of linear equations.

ii. Course Objectives

The objectives of this course are, to;

- teach techniques of interpolating data.
- carry out numerical differentiation and integration.
- solve non-linear equations and systems of linear equations using numerical schemes.
- write codes in Matlab or Python for simple numerical analysis algorithms.

iii. Learning outcomes

By the end of the course, the student should be able to

- derive schemes for different setups of numerical problems
- carry out numerical differentiation and integration
- solve non-linear equations and systems of linear equations using numerical techniques
- write codes for simple numerical analysis algorithms.

iv. Mode of delivery

1. Mostly lecture-oriented, but students can still interrupt the instructor and ask questions.
2. Students are encouraged to seek help outside the lecture room from the course instructor.
3. There will be bi-weekly assignments to be handed in the following week.
4. There will be at least two major homework assignments and two tests.
5. To be taught with ODeL provision

v. Prerequisites: None

vi. Indicative content

1. *Finite differences*: forward finite difference operators, backward finite difference operator, central difference operator, averaging operator, shift operator. [7 hours]
2. *Interpolation*: definition of interpolation, finite difference interpolation, finite difference tables, Newton's forward and backward difference interpolating

polynomial. Lagrange interpolation, linear, quadratic and higher degree Lagrange interpolating polynomials, error analysis in Lagrange interpolating polynomials, divided difference interpolation, Newton's divided difference interpolation and codes of interpolation. [8 hours]

3. *Numerical differentiation*: Why numerical integration, numerical differentiation using finite differences derivatives using Newton's forward formula, derivatives using Newton's backward difference formula. Error analysis in numerical differentiation. [8 hours]
4. *Numerical integration*: Trapezoidal rule, Simpson's rule, Analysis of errors in Trapezoidal and Simpson's rules. Computer codes for the algorithms of Trapezoidal and Simpson's rule. [6 hours]
5. *Numerical solution of non-linear equations*: Bisection method, Secant method, Fixed point/iteration/successive substitutions, Regular false method, Newton-Raphson method, Computer codes for the algorithms above. [8 hours]
6. *Numerical solution of a system of linear equations*: Direct methods, Gaussian elimination, Triangular decomposition, Cholesky's decomposition, Iterative techniques: Jacobi and Gauss-Seidel methods, Relaxation methods. Convergence analysis of the iterative methods. [8 hours]

vii. Assessment

- Course work (Tests, group assignments (take home assignments, case studies, individual / group projects)) 30%
- Final written exam: 70%

viii. Reading List

- Timothy Sauer, 2011, Numerical Analysis, 2nd Edition 2nd Edition, Pearson, ISBN-13 : 978-0321783677, ISBN-10 : 0321783670
- Brian Sutton, 2019, Numerical Analysis: Theory and Experiments, SIAM, ISBN:978-1-611975-69-7
- James V Lambers; Amber C Sumner, 2019, Explorations in numerical analysis, Singapore ; Hackensack, NJ : World Scientific Publishing Co. Pte. Ltd., ISBN: 9789813209961 9813209968 9789813209978 9813209976
- C.E. Froberg, 1994 Introduction to Numerical Analysis, Addison-Wesley.
- L.B. Richard, 1989, Numerical Analysis, Weber and Schmidt, Boston, Massachusetts.
- P.J. Oates, 1981, Numerical Analysis, Edward Arnold Publishing Ltd.
- J. M. Mango, Introduction to Numerical Analysis, IACE, Makerere University,

1.2.3 BSE1209 Object Oriented Programming I

Course Level	: Year I Semester II
Course Credit	: 4
Contact Hours	: 60

i. Course description

The course introduces concepts of object-oriented programming. The course shall not be bent on a single language of implementation, but rather on concepts. Languages such as Java, Python, Ruby and any other object-oriented programming language may be used to demonstrate the concepts. The course will not cover essentials of programming, including conditionals, iterative execution, and variable naming since they will already be covered in BSE1106(Problem Solving and Programming Concepts).

ii. Learning objectives

The objectives of the course include;

- To introduce the students to various conepts in object oriented programing
- To teach students basics to application development
- To provide students with definint context for object ortiented programming

iii. Llearning outcomes

Upon successful completion of this course, students will be able: -

- To differentiate between a class and object
- To create classes and instantiate objects
- To effectively use an IDE of choice to compile, debug and run object-oriented programs
- Defining and invoking methods
- To create and use packages
- Be able to differentiate between an interface and an abstract class and their applicability in object-oriented design.

iv. Indicative content

1. Classes and objects	(6 Hours)
2. Object instantiation	(6 Hours)
3. Arrays and Strings	(6 Hours)
4. Class and instance members (methods and data)	(6Hours)
5. Inheritance	(8 Hours)
6. Encapsulation	(8 Hours)
7. Polymorphism	(8 Hours)
8. Overloading and overriding	(6 Hours)
9. Abstract classes and Interfaces	(6 Hours)

v. Mode of delivery

The Topic adopts a student- centered learning involving the following:

1. Lecture presentations and guided discovery method

2. Classroom presentations

3. Group projects and learner-assisted discussions on assigned topics

4. To be taught with ODeL provision

vi. Assessment method

- Course work (Tests, group assignments (take home assignments, case studies, individual / group projects)) 30%
- Final written exam: 70%

vii. References

- Owsiak, T. (2017). Beginning C# 7 Hands-On–The Core Language. Packt Publishing Ltd.
- Jackson, W. (2016). Objects and Object-Oriented Programming: OOP Primer. In JSON Quick Syntax Reference (pp. 31-50). Apress, Berkeley, CA.

- Savitch, W. J., & Mock, K. (2015). Problem Solving with C++. Pearson.
- Burd, B. (2017). Beginning programming with Java for dummies. John Wiley & Sons.

1.2.4 IST1203 Data and Information Management I

Course Level : Year I Semester II
Course Credit : 4
Contact Hours : 60

i. Course Description

This course provides the students with an introduction to the core concepts in data and information management. It is centered around the core skills of identifying organizational information requirements, modeling them using conceptual data modeling techniques, converting the conceptual data models into relational data models and verifying its structural characteristics with normalization techniques, and implementing and utilizing a relational database using an industrial-strength database management system.

ii. Course Objectives

The objectives of this course are, to;

- Provide the students with systematic approaches to the design and implementation of databases
- Equip students with hands on experience and knowledge in developing databases and management of data.

iii. Course Learning Outcomes

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

1. Demonstrate an understanding of the role of databases and database management systems in managing organizational data and information in the context of enterprise systems; the historical development of database management systems and logical data models; the basics of how data is physically stored and accessed & the fundamentals of the basic file organization techniques.
2. Capture the information requirements for an enterprise domain using at least one conceptual data modeling technique (such as entity-relationship modeling).
3. Design high-quality relational databases
4. Demonstrate an understanding of the purpose and principles of normalizing a relational database structure.
5. Implement a relational database design using an industrial-strength database management system, including the principles of data type selection and indexing using the data definition, data manipulation, and data control language components of SQL Understand the basic mechanisms for accessing relational databases.

iv. Prerequisites: IST1101 Foundations of Information Systems and Technology

v. Indicative Content

This course will in the preliminary cover operations like requirements gathering and database planning. The course will also introduce students to developing application programs that talk to a database. These applications may be online or offline. The course will also include the following topics:

1. Database approach

(4 Hours)

2. Types of database management systems (4 Hours)

3. Basic file processing concepts and physical data storage concepts including file organizations techniques (4 Hours)
4. Database Development Life Cycle (8 Hours)
5. Conceptual data model (Entity-relationship model) (12 Hours)
6. Logical Database Design (Relational data model, database design, Mapping conceptual schema to a relational schema; Normalization) (12 Hours)
7. Physical database Design (Physical data model; Indexing; Datatypes) (4 Hours)

8. Database development (*Database Languages: SQL*) (12 Hours)

vi. Mode of delivery

Using text books, conference and journal publications, and online resources. The course will be delivered through lectures, tutorials and laboratory practicals.

To be taught with ODeL provision

Students will be offered a range of experiences that include:

- Lectures
- Online learning management systems(such as Moodle) taught with ODeL provision
- Class discussions and presentations
- Problem-based/case studies
- Project-based assignments that can be done in groups

vii. Mode of Assessment

- Course work (Tests, group assignments (take home assignments, case studies, individual / group projects)) 30%
- Final written exam: 70%

viii. Reading List

- Ma Feicheng, Marchionini Gary and Wu Dan, 2017, Data and Information Management, Volume 4 (2020): Issue 3 (Sep 2020): Special Issue: Data and Information Management under COVID-19 Global Pandemic, ISBN: 2543-9251
- Silberschatz, A. Korth, H.F. & Sudarshan. (2010). *Database* Systems Concepts (6th ed.). Publisher: McGraw-Hill.
- Gordon, K. (2007). Principles of Data Management: Facilitating Information Sharing.
- Connolly, T. & Begg, C. (2005). Database Systems: A Practical Approach to Design, Implementation, and Management (4th ed). Pearson Addison Wesley.
- William McKnight, 2014, Information Management, 1st Edition, Strategies for Gaining a Competitive Advantage with Data, Morgan Kaufmann, ISBN: 9780124080560

1.2.5 BSE1208 Introduction to Web Development

Course Level : Year I Semester II

Course Credit : 4

Contact Hours : 60

i. Course Description

The demand for usage of the web as the platform for business and service delivery, collaboration, social networking among others is increasing at a very terrific speed. Since the discovery of the World Wide Web, a number of technologies have been developed to support web development. In this course, we focus on exploring web technologies that can are used today to meet the demand of web services.

ii. Course Objectives

The aims of this course are to;

1. Explore web development tools and apply them in developing standard web applications
2. Equipe the student with practical skills for analyzing and designing simple web applications for organizations.

iii. Course Learning Outcomes

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

1. Demonstrate an understanding of the various web development tools and their applicability in creating web applications.
2. Analyse and design simple web applications for organizations.

iv. Indicative Content

The course focuses on exploring Web development technologies such as;

1. Web development IDEs	(10 Hours)
2. HTML5 (15 Hours)	
3. JavaScript and its variables such as NodeJs	(15 Hours)
4. Web content managenet systems	(10 Hours)
5. Web development frameworks	(10 Hours)

v. Mode of delivery

This will be done in form of web development projects by groups of students. Lectures shall also be used where the course instructor will guide and supervise the projects being done by the students.

To be taught with ODeL provision

vi. Mode of Assessment

- Course work (Tests, group assignments (take home assignments, case studies, individual / group projects)) 30%
- Final written exam: 70%

vii. Reading List

- Eric Matthes, 2019, Python Crash Course, 2nd Edition: A Hands-On, Project-Based Introduction to Programming, Illustrated, ISBN-10 : 1593279280
- Shenoy, A., 2014, Thinking in CSS, Publisher: Packt Publishing
- Fischer, W., 2016. CSS: Quick Start Guide – Effective Web Design (CSS, HTML, JavaScript, Programming), Publication Date: March 24, 2016
- Jörg Krause, 2017, Introducing Web Development, ISBN-10: 1484224981

1.3 Year I Recess

1.3.1 BSE1302 Software Engineering Practical skills Project I

Course Level: Year I Recess Term

Course Credit	: 5
Contact Hours	: 75

i. Course Description

This course provides students with a holistic hands-on experience in building software products in the C and web programming platforms in a team environmentims.

ii. Course Objectives

The objectives of this course are, to:

1. equip students with practical skills in software engineering from requirements solicitation to testing;
2. provide students with mentorship in writing software products in C and web programming environments;
3. give students an opportunity to build team-work and leadership skills needed in the workforce.

iii. Learning outcomes

Upon successful completion of the course, the student should:

1. Demonstrate mastery of a specific (C and web) programming language and development platform
2. Carry out requirement engineering for a particular project; and
3. Demonstrate ability to develop a mini application or systems with minimal difficulty.

iv. Mode of delivery

Lectures, practicals, group work, & tutorials

To be taught with ODeL provision

v. Prerequisites: C and web programming courses

vi. Indicative content

- A review of concepts for the programming tools to be used in the projects(such as C and Web development tools)
- Software development mini group/team projects using given programming languages already covered.

NB: Being practical hands-on recess work, the hours allocation to the content covered will depend on the project assigned to the students. However, this course has a total of 150 practical hours in recess of 10 weeks.

vii. Assessment

- Course work (Tests, group assignments (take home assignments, case studies, individual / group projects)) 30%
- Final written exam: 70%

viii. Reading List

- Teach Yourself C in 21 days, SAMS Publishing
- C Programming by Brian Kernighan and Denis Ritchie.
- Shenoy, A. (2014). Thinking in CSS, Publisher: Packt Publishing (April 14, 2014)
- Fischer, W. (2016). CSS: Quick Start Guide – Effective Web Design (CSS, HTML, JavaScript, Programming), Publication Date: March 24, 2016
- Tittel, E. & Noble, J. (2011). HTML, XHTML and CSS for Dummies, Publisher: For Dummies; 7 edition (January 11, 2011)
- Felke-Morris, T. (2010). Web Development and Design Foundations with XHTML, 5th Edition; Publisher: Pearson; 5 edition (March 6, 2010)

1.4 Year II Semester I

1.4.1 CSC2114 Artificial Intelligence

Course Level : Year II Semester I
Course Credit : 4
Contact Hours : 60

i. 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 artifcial 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.

ii. 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.

iii. Learning outcomes

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

1. Explain what constitutes "Artificial" Intelligence and how to identify systems with Artificial Intelligence.
2. Compare AI with human intelligence, and discuss its strengths and limitations and its application to complex and human-centered problems.
3. 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.
4. Apply the basic principles, models, and algorithms of AI to recognize, model, and solve real world problems that require intelligence.
5. Analyze the structures and algorithms of a selection of techniques related to searching, reasoning, machine learning, language processing, computer vision, image processing and robotics.

iv. **Mode of delivery**

Teaching and learning is implemented through Lectures, Tutorials sessions and Assignments/Quizzes. To be taught with ODeL provision. 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.

v. **Prerequisites:** BSE 1106 Problem Solving and Programming Concepts

vi. **Indicative content**

1. **Topic 1**: Introduction to Artificial intelligence (AI). (4 hours).
   The topic 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.
2. **Topic 2**: Planning- uninformed search (8 hours).
   The topic 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.
3. **Topic 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.
4. **Topic 4:** - Constraint Satisfaction Problems(CSPs) (12hours).
   The topics 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 upthe backtracking search: filtering, Arc Consistency, ordering, exploring the constraint graph structure, Iterative Algorithms for CSPs such as minimum conict algorithm.
5. **Topic 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, multi-agent utilities and Maximum expected utility (MEU).
6. **Topic 6:** Non-Deterministic Search and Reinforcement learning(10 hours). The topics provides an introduction to Markov Decision Processes and Policies as well as an introduction Reinforcement learning.
7. **Topic 7**: Bayesian Network (10 hours).
   The Topic introduces the Bayesian network, Representation , Independence and Inference.

vii. **Assessment**

- Course work (Tests, group assignments (take home assignments, case studies, individual / group projects)) 30%

- Final written exam: 70%

viii. **Reading List**

- Russell Stuart Jonathan and Norvig Peter. Articial Intelligence: A Modern Approach. 3rd Edition, Prentice Hall, 2010.
- Philip C Jackson. Introduction to Articial Intelligence. Second enlarged Edition.

1.4.2 CSC2100 Data Structures and Algorithms

Course Level: Year II Semester I
Course Credit : 4
Contact Hours : 60

i. **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.

ii. **Course Objectives**

The objectives of this course are to educate the students in:

1. 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.
2. Define and explain advanced data types such as stacks, queues, lists, trees and graphs; write programs using them.
3. Define, discuss and explain the main algorithms and techniques (such as sorting, searching, hashing, traversal and recursion) and write programs using these algorithms.

iii. **Learning outcomes**

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

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

iv. **Mode of delivery**

The teaching pattern is by lectures and tutorials, practical lab work, group discussions and class presentations.
To be taught with ODeL provision

v. **Prerequisites:** BSE 1106 Problem Solving and Programming Concepts

vi. **Indicative content**

1. **Topic 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.
2. **Topic 2:** Recursive data structures (14 CH) recursion; lists, stacks, queues, dequeues, linked lists and their implementation; Trees: binary trees, balanced trees.

3. **Topic 3**: Sorting Algorithms (12 CH) Simple: bubble sort, insertion sort, selection sort; Fast: Mergesort, Heap sort, Quick sort.
4. **Topic 4:** Storing and Searching (8 CH) hash tables, binary search **trees**, AVL trees.
5. **Topic 5:** Graph traversals (8 CH) Breadth-First Search and Depth-First Search; shortest path, spanning trees.
6. **Topic 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.
7. **Topic 7:** Algorithm design (4 CH) Dynamic algorithms, Divide-and-conquer algorithms, Greedy algorithms.

vii. **Assessment**

- Course work (Tests, group assignments (take home assignments, case studies, individual / group projects)) 30%
- Final written exam: 70%

viii. **Reading List**

- Mark Allen Weiss Data Structures and Algorithm Analysis in C++, 4th Edition, 2014
- Michael T. Goodrich and Roberto Tamassia and Michael H. Goldwasser. Data Structures and Algorithms in Python, 2013.
- Alfred V. Aho and John E. Hopcroft and Jeffrey D. Ullman. Data Structures and Algorithms. Addison-Wesley, 1983.
- Thomas H. Cormen and Charles E. Leiserson and Ronald L. Rivest and Clifford Stein. Introduction to Algorithms, Second Edition, 2001.

1.4.3 BSE2106 Computer Networks

Course Level: Year II Semester I
Course Credit : 4
Contact Hours : 60

i. **Course Description**

This is an introductory course in computer networks This course provides a broad overview of computer networking, covering application layer, transport layer, network layer, and link layers. It covers basic concepts in computer networking as well as the prominent Internet protocols. It also ensures that students practically appreciate them.

ii. **Course Objectives**

The course objectives are, to:

1. Familiarise students with the terminology and concepts of the OSI reference model and the TCP/IP reference model protocols.
2. Explore performance issues in local area networks and wide area networks, and wireless networks;
3. Give students knowledge and skills in network tools, practical network set up and network trouble shooting

iii. **Learning outcomes**

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

1. Master the terminology and concepts of the OSI reference model and the TCP/IP reference model protocols;
2. Appreciate performance issues in local area networks and wide area networks, and wireless networks;
3. Demonstrate knowledge of network tools.

4. Practically set up a network, trouble shoot

iv. Mode of delivery

This course will be delivered through class lectures, discussions and laboratory sessions. In addition, students shall be assigned with practical project in which they shall demonstrate practical knowledge of network setup, configuration and troubleshooting. To be taught with ODeL provision

v. Indicative content

1. Introduction to Networks: definition, advantages, types, configurations; network topologies, types of networks (8 Hours)
2. The OSI/ISO reference model; Transmission media: magnetic media, twisted pair, coaxial, fiber-optics; (10 Hours)
3. Packet switching, delay and loss concepts, physical media, protocol layering, (10 Hours)
4. Application layer: Web, E-mail, DNS, FTP (8 Hours)
5. Transport layer: how UDP and TCP (8 Hours)
6. Network layer and routing routing protocol basics, RIP , IPv4 and IPv6, addressing and CIDR (8 Hours)
7. Computer Network security basics (8 Hours)

vi. Assessment Method

- Course work (Tests, group assignments (take home assignments, case studies, individual / group projects)) 30%
- Final written exam: 70%

vii. Reading List

- James F. Kurose and Keith W. Ross. Computer Networking - A Top Down Approach Featuring the Internet, 5th edition, Addison-Wesley, ISBN 0-321-22735-2.
- Andrew S. Tanenbaum Computer Network (4th Edition) Prentice Hall
- W. Richard Stevens, TCP/IP Illustrated, Vol. 1: The Protocols, Addison-Wesley Professional; US ed edition (December 31, 1993)
- Eric Rescorla, SSL and TLS: Designing and Building Secure Systems, Addison-Wesley Professional; 1 edition (October 27, 2000)

1.4.4 BSE2105 Formal Methods

Course Level: Year II Semester I
Course Credit : 3
Contact Hours : 45

i. Course Description

The course introduces students to application of mathematical techniques to reason and think about computations.

ii. Course Objectives

The objectives of this course are, to;

- Enable the student have an appreciation of the professional need to establish formal properties of software;
- Make the students have a belief that formal specifications can improve the quality of software.

iii. Learning Outcomes

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

- Use the Event-B notation to develop and prove software specifications
- Install an Event-B Toolkit on a Unix/Linux/Windows platform

- Write basic Event-B specifications;
- Refine and extend more advanced Event-B specifications.

iv. Mode of delivery

Lecture notes, student presentations, short exploratory assignments
To be taught with ODeL provision

v. Indicative Content

Topics to be covered include:

1. Review of set theory (3 Hours)
2. The predicate calculus (3 Hours)
3. Relations (3 Hours)
4. Relational algebra and formal specification concepts (4 Hours)
5. Algebraic and model based specifications (4 Hours)
6. The role of formal specifications in software engineering(4 Hours)
7. The Event-B notation(6 Hours)
8. Data and algorithm design(6 Hours)
9. Data and operation refinement(4 Hours)
10. Proofs of correctness(4 Hours)
11. Proof obligations. (4 Hours)

vi. Assessment Method

- Course work (Tests, group assignments (take home assignments, case studies, individual / group projects)) 30%
- Final written exam: 70%

vii. Reading List

- Jean-Raymond Abrial, The B-Book: Assigning Programs Meanings, Cambridge University Press, 1996, ISBN 0-521-49619-5
- John Wordworth, Software Engineering with B, Addison Wesley Longman, 1996, ISBN 0-201-40356-0
- Kevin Lano Specifications in B: An Introduction using B Toolkit, World Scientific Publishing Company, Imperial College Press, ISBN 1-86094-008-0
- Jean Francois Monin, Michael Gerard Hinchey, Understanding Formal Methods, Springer, 2003
- Jonathan Bowen, Formal Specification and Documentation using Z: A Case Study Approach, Centre for Applied Formal Methods, London South Bank University, Thomson Publishing , 2003

1.4.5 BSE2107 Object Oriented Programming II

Course Level: Year II Semester I
Course Credit : 4
Contact Hours : 60

i. Course description

The course builds on the concepts of object-oriented programming covered in Object Oriented Programming I to further explore more object oriented programming concepts and their applicability. The course shall not be bent on a single language of implementation, but rather on the applicability of concepts. Languages such as Java, Python, Ruby and any other object-oriented programming language may be used to demonstrate the concepts.

ii. Learning objectives/ learning outcomes

Upon successful completion of this course, students will be able:

- To demonstrate and apply the object oriented programming principles in building applications

- To work with current frameworks for developing applications using object oriented programming paradigm.

iii. Indicative content

• Advanced concepts in inherantance and polymorphism	(15 Hours)
• Object serialization and deserialization	(15 Hours)
• Persistence	(10 Hours)
• Working with GUI	(10 Hours)
• Example frameworks for application development	(10 Hours)

iv. Mode of delivery

The course adopts a student- centered learning involving the following:

- Lecture presentations and guided discovery method
- Classroom presentations
- Group projects and learner-assisted discussions on assigned topics

To be taught with ODeL provision

v. Assessment method

- Course work (Tests, group assignments (take home assignments, case studies, individual / group projects)) 30%
- Final written exam: 70%

vi. References

- Owsiak, T. (2017). Beginning C# 7 Hands-On–The Core Language. Packt Publishing Ltd.
- Jackson, W. (2016). Objects and Object-Oriented Programming: OOP Primer. In JSON Quick Syntax Reference (pp. 31-50). Apress, Berkeley, CA.
- Savitch, W. J., & Mock, K. (2015). Problem Solving with C++. Pearson.
- Burd, B. (2017). Beginning programming with Java for dummies. John Wiley & Sons.

1.5 Year II Semester II

1.5.1 CSC2200 Operating Systems

Course Level: Year II Semester II
Course Credit : 4
Contact Hours : 60

i. 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.

ii. Course Objectives

The objectives of this course are, to;

- 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.

iii. 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-o_s inherent in OS implementation

iv. Indicative Content

1.	Introduction to Operating Systems	- 4 hours
2.	Processes, Threads and Dispatching, Concurrent Threads	- 8 hours
3.	Locks and Condition Variables, Implementing Locks	- 8 hours
4.	CPU Scheduling	- 6 hours
5.	Deadlocks	- 4 hours
6.	Linkers and Dynamic Linking	- 6 hours
7.	Dynamic Storage Management	- 4 hours
8.	Memory Management, Virtual Memory, Flash Memory	- 8 hours
9.	Demand Paging	- 4 hours
10.	Virtual Machines	- 4 hours
11.	File System Crash Recovery	- 4 hours

v. Mode of delivery

The teaching pattern is by lectures, tutorials, practical lab work, group discussion and class presentations. To be taught with ODeL provision

vi. Mode of Assessment

- Course work (Tests, group assignments (take home assignments, case studies, individual / group projects)) 30%
- Final written exam: 70%

vii. Reading List

- Operating systems: principles and practice, Anderson, Thomas and Dahlin, Michael, 2nd Edition, Recursive books, 2014
- Jeff Geerling, 2020, Ansible for DevOps: Server and configuration management for humans , Midwestern Mac, LLC, ISBN-10 : 0986393428,
- Daniel J. Barrett, 2016, Linux Pocket Guide: Essential Commands 3rd Edition, O'Reilly Media, ISBN-10 : 1491927577,ISBN-13 : 978-1491927571
- Jaosn Cannon, 2015, Shell Scripting: How to Automate Command Line Tasks Using Bash Scripting and Shell, ISBN-10 : 151738043X
- Mark G. Sobell, 2015, A Practical Guide to Ubuntu Linux 4th Edition, earson P T R; ISBN-10 : 0133927318, ISBN-13 : 978-0133927313
- Mike Meyers, 2019, CompTIA A+ Certification All-in-One Exam Guide, Tenth Edition (Exams 220-1001 & 220-1002) 10th Edition, ISBN-13 : 978-1260454031, McGraw-Hill Education;
- James Jordan, 2020, WINDOWS 10 FOR SENIORS 2020/2021: The

Complete Microsoft Windows 10 Guide for Senior Technophobe with Latest Shortcuts, Tips & Tricks, ISBN-13 : 979-8562469175, Independently published

1.5.2 BSE2207 Emerging Web Development Technologies

Course Level : Year II Semester II
Course Credit : 4
Contact Hours : 60

i. Course Description

The course shall cover new trends in the web development field. This shall include a mention of the technologies as well as demonstrating them practically. The course shall be more on encouraging students to explore the new technologies through providing weekly practical assignments, which shall be assessed.

ii. Course Objectives

The objectives of the course are to enable the students to understand;

1. AI and bots and how to implement them in web development.
2. Container management platforms and how to implement them.
3. How to develop progressing web applications and develop a sample application
4. Web development frameworks and use one of such frameworks to develop a web application

iii. Learning Outcomes

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

- Know AI and bots and how to implement them using a method of choice
- Know what container management platforms are and how to implement them
- Know progressing web applications and develop sample applications
- Identify and use any web framework of choice to develop applications

iv. Indicative content

1. AI and bots with references from examples such as Microsoft Bot frameworks, facebook Bots etc
2. Container management platforms like Docker
3. Progressing web applications such as angular, react, polymer
4. API-first e.g swagger
5. Google Amp
6. JavaScript variants like typescript, nodejs
7. New web development frameworks

v.Mode of delivery

The course adopts a student- centered learning involving the following:

1. Lecture presentations and guided discovery method
2. Classroom presentations
3. Group projects and learner-assisted discussions on assigned topics

The course is to be taught with ODeL provision

vi. Assessment method

- Course work (Tests, group assignments (take home assignments, case studies, individual / group projects)) 30%
- Final written exam: 70%

vii. References

- Krochmalski, J. (2017). Docker and Kubernetes for Java Developers. Packt Publishing Ltd.
- Sayfan, G. (2017). Mastering Kubernetes. Packt Publishing Ltd.
- Sheppard, D., & Sheppard, D. (2017). Beginning Progressive Web App Development. Apress
- Rozga, S. (2018). Practical bot development: designing and building bots with Node. js and Microsoft Bot Framework. Apress

1.5.3 BSE2208 Requirements Engineering

Course Level	: Year II Semester II
Course Credit	: 3
Contact Hours	: 45

i. Course Description

The course will discuss concepts for systematically establishing, defining and managing the requirements for large, complex, changing and software-intensive systems, from technical, organizational and management perspectives. The course will consider the past, present and future paradigms and methodologies in requirements engineering. The course will cover informal, semi-formal and formal approaches, while striking a balance between theory and practice. The course will involve building models of both requirements engineering process and requirements engineering product, concerning both functional and non-functional goals/requirements/specifications, using a systematic decision-making process.

ii. Course Objectives

The objectives of this course is to enable the students to understand;

1. the need for requirements for large-scale systems
2. the stakeholders involved in requirements engineering
3. requirements engineering processes
4. models of requirements
5. functional requirements
6. non-functional requirements
7. scenario analysis and
8. object-oriented and goal-oriented requirements engineering.

iii. Learning outcomes

On completion of the course, the student will be able to;

1. understand the basics of Requirements Engineering
2. prepare for, and undertake the requirements elicitation tasks
3. analyse client needs
4. create models of requirements using a variety of notations and techniques
5. prepare software requirements specifications using an industry standard
6. prepare for, and undertake formal specification reviews.

iv. Mode of delivery

The teaching and learning approaches will combine classroom lectures with theories and discussion of case studies in groups. Take home assignments / coursework will be administered. To be taught with ODeL provision

v. Indicative Content

1. **1.** Introduction to Requirements Engineering (4 Hours)

2. Requirements Engineering Processes	(6 Hours)
3. Requirements Models	(4 Hours)
4. Requirements Analysis, and Modeling	(4 Hours)
5. Requirements Elicitation	(4 Hours)
6. Scenario Analysis	(3 Hours)
7. Modeling Enterprises	(3 Hours)
8. Structured Analysis	(4 Hours)
9. Goal-Oriented Requirements Engineering	(3 Hours)
10. Specifications & Validation	(4 Hours)
11. Managing Change and Inconsistency	(6 Hours)

vi. Assessments Method

- Course work (Tests, group assignments (take home assignments, case studies, individual / group projects)) 30%
- Final written exam: 70%

vii. References

- G. Kotonya and I. Sommerville, 1998, Requirements Engineering: Processes and Techniques, John Wiley Sons
- Hull, Elizabeth, Jackson, Ken, Dick, Jeremy , 2011, Requirements Engineering, Springer-Verlag London, DOI: 10.1007/978-1-84996-405-0, ISBN: 978-1-4471-5818-9
- Gerardus Blokdyk, 2019, Requirements engineering A Complete Guide - 2019 Edition 5starcooks, ISBN-10 : 0655549072, ISBN-13 : 978-0655549079

1.5.4 BSE2209 Mobile Programming Project

Course Level : Year II Semester II
Course Credit : 4
Contact Hours : 60

i. Course Description

This course introduces the student to development of applications that can run on various mobile platforms like Android phones and iPhones. The focus is on providing students with a holistic hands-on experience in building software products on the mobile platform in a team environment. A number of mobile applications for solving day-to-day business and industrial problems are developed for various mobile platforms in this course.

ii. Course Objectives

These are, to;

1. Explore the various tools and techniques for developing mobile applications and apply them to develop mobile applications for various platforms.
2. Equip students with practical skills in developing mobile applications for computing resource constrained platforms
3. Empower students to identify gaps in industry for mobile programmes

iii. Learning outcomes

Upon successful completion of the course, the student should:

1. Practically demonstrate mastery of the tools and techniques for developing mobile applications that can run on various mobile platforms.
2. Practically demonstrate the skills in developing mobile applications for computing resource constrined platforms.
3. Be able to technically demonstrate services in mobile applications

iv. Mode of delivery

Lectures, practicals, group work, & tutorials
The course is to be taught with ODeL provision

v. Indicative content

1. An introduction to programming language for developing mobile applications - should be more of student independent study. (6 Hours)

Choice to depend on the programming languages already covered in the course

2. Mobile application platforms (6 Hours)
3. C# and libraries (8 Hours)
4. .NET and libraries (8 Hours)
5. IDEs for developing mobile applications (such as Android Studio) (12 Hours)
6. Frameworks for mobile applications development (20 Hours)

-Swiftic
-Native scripts
-React Native
-Xamarin
-JQuery mobile
- Mobile Angular UI

vi. Assessment Method

- Course work (Tests, group assignments (take home assignments, case studies, individual / group projects)) 30%
- Final written exam: 70%

vii. Reading List

- Eric Matthes, 2019 Python Crash Course, 2nd Edition: A Hands-On, Project-Based Introduction to Programming Paperback – Illustrated, ISBN-10 : 1593279280
- Mark J. Price 2020, C# 9 and .NET 5 – Modern Cross-Platform Development: Build intelligent apps, websites, and services with Blazor, ASP.NET Core, and Entity Framework Core using Visual Studio Code, 5th Edition, ISBN-10 : 180056810
- Shaw Zed A., 2013, Learn Python the Hard Way: A Very Simple Introduction to the Terrifyingly Beautiful World of Computers and Code (Zed Shaw's Hard Way Series) 3rd Edition, Addison-Wesley Professional; ASIN : B00FGUS948
- Steve Karam, and Robert Freeman, 2018, Easy Oracle Jumpstart: Oracle Database Management Concepts and Administration, ISBN 0-9759135-5-7
- Steven Feuerstein, Bill Pribyl and Chip Dawes 2007,Oracle PL/SQL Language Pocket Reference, by Fourth Edition, OReilly, ISBN 10: 0-596-51404-2

1.5.5 BSE2206 Data Communication

Course Level : Year II Semester II
Course Credit : 4
Contact Hours : 60

i. Course Description

This is a theoretical course that covers the fundamentals of data communication, formatting and transmission of digital information over various media.

ii. Course Objectives

The objectives of this course are to;

1. provide a good understanding of the electrical characteristics of digital signals and the basic methods of data transmission.
2. introduce the concept of communication protocols and give an overview of Data Communication Standards.
3. introduce the area of data and compression, with emphasis on the range of communication protocols utilized.
4. explore the concept of Open Systems, giving an overview of Transport and Application Support Protocols.

iii. Learning Outcomes

At the end of this course, students should;

1. understand basic functions on which modern communication systems are built
2. know how packets find their way through the Internet, and how congestion is avoided
3. know how wireless communication works
4. be able to develop programs that efficiently communicate over a network
5. to improve C programming skills
6. have the required basis for working in this area

iv. Mode of delivery

This course will be delivered through class lectures, discussions. Simulations will also be used to explain abstract concepts.

The course is to be taught with ODeL provision

v. Indicative content

1. Introduction to communication (8 Hours)
2. Digital versus Analog transmission; (8 Hours)
3. Modems (4 Hours)
4. Transmission media: magnetic media, twisted pair, coaxial, fiber-optics; (8 Hours)
5. Data encoding: straight, Manchester, differential Manchester, satellite; (8 Hours)
6. Modulation and their standards, codes and pulse code modulation;
7. Integrated Services Digital Networks (ISDN); (8 Hours)
8. Network Access Protocols; Passive versus dynamic allocation; (8 Hours)
9. LAN standards:802.3 (Ethernet),802.4 (token bus), 802.5 (token ring) (8 Hours)

vi. Assessment Method

- Course work (Tests, group assignments (take home assignments, case studies, individual / group projects)) 30%
- Final written exam: 70%

vii. Reading List

- Jerry FitzGerald, Alan Dennis, Alexandra Durcikova, 2017, Business Data Communications and Networking, Wiley, ISBN 1119368898, 9781119368892
- Behrouz A. Forouzan, 2012, Data Communications and Networking, Global Edition, McGraw-Hill Education, ISBN-10 : 0071315861, ISBN-13 : 978-0071315869

- Kurose, J.F & Ross, K.W, 2007Computer Networking: A Top Down Approach, 4th edn, Addison-Wesley
- David Stamper 2003,, Business Data Communications, 6th Edition, Prentice Hall.
- Fred Halsall, 1998 *Data Communications, Computer Networks, and Open Systems, 4th Edition*, Addison-Wesley.

1.6 Year II Recess

1.6.1 BSE 2302 Software Engineering Practical Skills Project II

Course Level	: Year II Recess
Course Credit	: 5
Contact Hours	: 75

i. **Course Description**

This course builds on the knowledge and skills acquired in BSE2209 Mobile Programming Project to enhance the student's mobile applications development skills. The course further focuses on integration of the mobile plaform applications with applications running on other plaforms such as the web. Students are expected to work on development projects in a team environment.

ii. **Course Objectives**

The objectives of this course are, to:

1. Enhance practical skills of the student in development of mobile applications and other applications that can integrate with the mobile platform(such as the web).
2. Provide students with mentorship in writing software products for mobile devices;
3. Give students an opportunity to build team-work and leadership skills needed in the workforce.

iii. **Learning outcomes**

Upon successful completion of the course, the student should:

1. Demonstrate mastery of mobile applications development and integration with other computing platforms
2. Be able to carry out requirement specification for a particular mobile programming project and develop such a specified software project.
3. Demonstrate ability to work in teams to solve a software development problem.

iv. **Mode of delivery**

Lectures, practicals, group work, & tutorials

The course is to be taught with ODeL provision

v. **Indicative content**

1. Programming languages and IDEs for mobile applications development
2. Frameworks for mobile applications
3. Integrating mobile apps with other platforms(such as web where the mobile app interacts with a web server).

NB: Being practical hands-on recess work, the hours allocation to the content covered will depend on the project assigned to the students. However, this course has a total of 150 practical hours(75 Contact Hours) in recess of 10 weeks

vi. **Assessment Method**

- Course work (Tests, group assignments (take home assignments, case studies, individual / group projects)) 30%

- Final written exam: 70%

vii. Reading List

- Martin Fowler 2018, Refactoring: Improving the Design of Existing Code (2nd Edition) (Addison-Wesley Signature Series (Fowler)) 2nd Edition, ISBN-10 : 0134757599
- Cory Berg 2015, Software++: Must-Have Skills for Software Engineers Kindle Edition, ISBN : 1514629348

1.7 Year III Semester I

1.7.1 BSE3110 Object Oriented Analysis and Design

Course Level	: Year III Semester I
Course Credit	: 4
Contact Hours	: 60

i. Course Description

This course introduces the fundamental principles of object oriented approaches to modeling software requirements and design. In this course, the students will learn how to produce detailed object models and designs from system requirements; use the modeling concepts provided by UML; identify use cases and expand into full behavioral designs; expand the analysis into a design ready for implementation and construct designs that are reliable. The course begins with an overview of the object oriented analysis and design. The following figure shows the flow of the course.

ii. Course Objectives

The objectives of this course are to enable the student to:

1. Apply an iterative process such as the Unified Process.
2. Analyze software requirements and document these requirements using Use Cases.
3. Perform software analysis and record the results using UML notation.
4. Perform software design and record the results using UML notation.
5. Apply object-oriented patterns

iii. Learning outcomes

At the end of the course, each student should be able to;

1. Design and implementation of programs
2. Formulate and solve problems in computing.
3. Understand design and performance requirements of software systems.
4. Apply sound principles to the synthesis and analysis of computer systems
5. Engage in lifelong learning and expect to embrace change
6. Communicate effectively and think critically and creatively, both independently and with others
7. Be aware of social and ethical issues of computers in society

iv. Mode of delivery

The teaching and learning approaches will combine classroom lectures with theories and discussion of case studies in groups. Take home assignments / coursework will be administered.

The course is to be taught with ODeL provision

v. Indicative content

1. Principles of Object Technology	(6 Hours)
2. Project Organization and Communication	(6 Hours)
3. Fundamentals of Visual Modeling with UML: Business Modeling.	(6 Hours)
4. Object-Oriented Analysis with UML.	(8 Hours)
5. Object-Oriented Design with UML (Part 1).	(8 Hours)
6. Object-Oriented Design with UML (Part 2).	(8 Hours)
7. Dealing with Complexity	(6 Hours)
8. Mapping Models to Code	(6 Hours)
9. Testing	(6 Hours)

vi. Assessment Method

- Course work (Tests, group assignments (take home assignments, case studies, individual / group projects)) 30%
- Final written exam: 70%

vii. Reading List

- Krishna Sankar P, Shangaranarayanee N P, 2019 "Object Oriented Analysis and Design for Engineering Studies", First Ed., A.R.S Publications, Chennai
- Bernd Bruegge and Allen H. Dutoit. Object-Oriented Software Engineering, Using UML, Patterns, and Java, 3rd Edition, Prentice-Hall, 2010, ISBN-10: 0136066836.
- Craig Larman, Applying UML and Patterns - An Introduction to Object-Oriented Analysis and Design and Iterative Development, Prentice Hall, 2004, ISBN: 0-13-148906-2. Resources

1.7.2 BSE3111 Embedded Systems I

Course Level	: Year III Semester I
Course Credit	: 4
Contact Hours	: 60

i. Course Description

The course covers introductory aspects of embedded systems. It will begin by defining embedded systems and identifying the architectures. The microcontrollers being used will be industry standard devices. C programing concepts specific to microcontrollers will be covered. The principles of interfacing to the microcontroller peripherals and their applications will be considered. That shall include an introduction to motors and sensors. These techniques will be brought together in ‘real world’ systems design during prototyping with real hardware.

ii. Course Objectives

The objectives of this course are, to;

1. Equip students with a comprehensive knowledge of embedded systems
2. Enable the students evaluate and analyse appropriate C programming techniques as applied to microcontrollers

3. Enable students understand and critically evaluate microcontrollers and their architectures.
4. Enable students to construct and critically evaluate interfacing syste,ms for use in embedded systems
5. Enable students to construct a prototype of an embedded system for an application of choice
6. Equip students with ability to interpret embedded device datasheets.

iii. Learning Outcomes

On completion of this course students will be able to;

1. Demonstrate a comprehensive knowledge of and be able to make effective use of embedded systems
2. Evaluate and analyse appropriate C programming techniques as applied to microcontrollers
3. Understand and critically evaluate microcontrollers and their architectures;
4. Construct and critically evaluate interfacing systems for use in embedded systems
5. Construct a prototype of an embedded system for an application of choice
6. Demonstrate comprehensive knowledge in interpreting embedded device datasheets

iv. Indicative content

1.	Embedded systems architecture	(6 Hours)
2.	Digital logic for embedded computing	(6 Hours)
3.	Electronics principles for embedded systems	(6 Hours)
4.	Schematic design and embedded systems simulation	(6 Hours)
5.	Case study of a micro-controller design	(6 Hours)
6.	Input/output peripherals interfacing to microcontroller	(6 Hours)
7.	C programming and tools for firmware / embedded system design	(6 Hours)
8.	Sensors, data acquisition and interfaces	(4 Hours)
9.	Motors	(4 Hours)
10.	Serial communication	(4 Hours)
11.	Prototyping with board of choice	(6 Hours)

v. Mode of delivery

1. Teaching, to establish a good fundamental understanding of the areas covered

2. Formal face-to-face lectures, to provide a focus on the core analytical material in the course, together with qualitative, alternative explanations to aid understanding;

3. Practical demonstrations to support the formal lecture material and also provid practical construction, measurement and debugging skills

4. The course is to be taught with ODeL provision

vi. Assessment Method

- Course work (Tests, group assignments (take home assignments, case studies, individual / group projects)) 30%
- Final written exam: 70%

vii. References

- Naimi, Sepehr, Sarmad Naimi, and Muhammad Ali Mazidi. "The AVR Microcontroller and Embedded Systems Using Assembly and C: Using Arduino Uno and Atmel Studio." (2017).
- Sanchez, Julio, and Maria P. Canton. Embedded systems circuits and programming. CRC Press, 2017.

1.7.3 BSE3104 Software Metrics

Course Level : Year III Semester I
Course Credit : 3
Contact Hours : 45

i. Course Description

This course introduces students to measurement theory in software, looking at measurement techniques, attributes in both processes and products.

ii. Course Objectives

The objectives of this course are to enable the student to;

1. Identify software metrics and the foundations of measurement theory and models of software engineering measurement.
2. Come up with software product metrics, software process metrics and measuring management mechanisms

iii. Learning outcomes

Upon successful completion of this course students shall be able to:

1. Describe software metrics and their application in software systems
2. Tell the foundations of measurement theory and models of software engineering measurement
3. Appreciate software products metrics, software process metrics and measuring management.

iv. Teaching & Learning patterns

Lectures, class presentations, assignments

The course is to be taught with ODeL provision

v. Indicative content

1. Measurement: what is it and why do it?	(3 Hours)
2. The basic of measurement	(3 Hours)
3. A goal-based framework for software measurement	(3 Hours)
4. Empirical investigation	(3 Hours)
5. Software-metrics data collection	(3 Hours)
6. Analyzing software-measurement data	(4 Hours)
7. Measuring internal product attributes: size	(4 Hours)
8. Measuring internal product attributes: structure	(4 Hours)
9. Measuring external product attributes	(4 Hours)
10. Software reliability: measurement and prediction	(3 Hours)
11. Resource measurement: productivity, teams, and tools(3 Hours)	
12. Making process predictions	(3 Hours)
13. Planning a measurement program	(3 Hours)

14. Measurement in practice	(3 Hours)

vi. Assessment Method

- Course work (Tests, group assignments (take home assignments, case studies, individual / group projects)) 30%
- Final written exam: 70%

vii. Reading List

- Norman Fenton, James Bieman, Taylor & Francis, Software Metrics: A Rigorous and Practical Approach, (3rd ed. illustrated, revised) (652p.), 2013. ISBN: 1439838224, 9781439838228.
- Stephen H. Kan Metrics and Models in Software Quality Engineering, , 2nd ed. (560 p.), Addison- Wesley Professional (2002). ISBN:0201729156.
- John C. Munson Software Engineering Measurement, , Auerbach Publications, 2003 (443 pages) ISBN:0849315034
- BA Kitchenham Software Metrics: Measurement for Software Process Improvement, Blackwell Pub, 1996.ISBN: 1855548208.

1.7.4 CSC3110 User Interface Design

Course Level	: Year III Semester I
Course Credit	: 4
Contact Hours	: 60

i. Course Description

The course introduces the principles of user interface development, focusing on design, implementation and evaluation.

ii. Course Objectives

The objectives of this course are, to;

1. Enable the student to develop efficient, flexible and interactive User Interfaces(UI) Provide ability to identifying system users, the tasks they want to carry out and the environment in which they will be working.
2. Equip the student with practical skills in creating conceptual designs, designing various kinds of UI, in particular graphical user interfaces (GUIs) and web sites and evaluating UIs.

iii. Learning Outcomes

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

1. Develop efficient, flexible and interactive user interfaces
2. Create conceptual designs, and other kinds of GUIs and evaluate such designs.

iv. Mode of delivery

The teaching pattern is by lectures, lab sessions and projects. The course is to be taught with ODeL provision.

v. Indicative content

1. Usability	(2 Hours)
2. User-Centered Design	(3 Hours)
3. UI Software Architecture	(3 Hours)
4. Human Capabilities	(3 Hours)
5. Output Models	(2 Hours)

6. Conceptual Models and Metaphors	(3 Hours)
7. Input Models	(2 Hours)
8. Design Principles	(3 Hours)
9. Paper Prototyping	(3 Hours)
10. Constraints and Layouts	(3 Hours)
11. Graphic Design	(3 Hours)
12. Computer Prototyping	(3 Hours)
13. Heuristic Evaluation	(3 Hours)
14. User Testing	(3 Hours)
15. Experiment Design	(3 Hours)
16. Experiment Analysis	(3 Hours)

vi. Assessment Method

- Course work (Tests, group assignments (take home assignments, case studies, individual / group projects)) 30%
- Final written exam: 70%

vii. Reading List

- Gerardus Blokdyk, 2020, UX UI Design A Complete Guide - 2020 Edition, ISBN: 9780655967439
- Norman, D. A. 1990, The Design of Everyday Things. New York, NY: Doubleday,. ISBN:0385267746.
- Nielsen, J. 1994, Usability Engineering. Burlington, MA: Academic Press, ISBN: 0125184069.
- Mullet, K., and D. Sano. 1994 Designing Visual Interfaces: Communication oriented techniques Prentice Hall. ISBN: 0133033899.
- Steve Krug, 2005 Don't Make Me Think: A Common Sense Approach to Web Usability, 2nd Edition, New Riders; 2nd edition
- Jef Raskin, 2000m The Humane Interface: New Directions for Designing Interactive Systems, Addison- Wesley Professional

1.7.5 BSE3105 Software Evolution

Course Level : Year III Semester I
Course Credit : 3
Contact Hours : 45

i. Course Description

This course introduces the student to evolution of systems. It focuses on issues in maintaining of legacy systems as well as configuration management practices in software industry.

ii. Course Objectives

The objectives of this course are, to;

- Teach students the process of software changes and how to manage them in an organization
- Enable the student appreciate the importance of legacy systems in organizatipons and the approaches used to ensure that such old systems remain relevant (in terms of quality and business value) to the success of the

organization.

- To make student understand how systems evolve and how to manage their evolution.

iii. Learning outcomes

At the end of this course, the student should;

- Demonstrate knowledge of different processes of software change and how they are implementated during software maintentance
- Undeestand the structure of legacy systems in organizations and how important they are to the success of the orgaization's business processes.

iv. Mode of delivery

In this course a combination of lectures and practical software projects will be used. The course is to be taught with ODeL provision

v. Indicative content

1. Introduction to Software evolution	(6 Hours)
2. Managing software change	(9 Hours)
3. Software Re-engineering	(9 Hours)
4. Evolution of legacy system	(9 Hours)
5. Configuration management	(12 Hours)

vi. Assessment Method

- Course work (Tests, group assignments (take home assignments, case studies, individual / group projects)) 30%
- Final written exam: 70%

vii. Reading List

- Reussner, R., Goedicke, M., Hasselbring, W., Vogel-Heuser, B., Keim, J., Märtin, L., 2019, Managed Software Evolution, DOI 10.1007/978-3-030-13499-0
- Ian Sommerville, 2016 Software Engineering, 10th Edition, University of St Andrews, Scotland, Pearson
- Priyadarshi Tripathy, Kshirasagar Naik 2014, Software Evolution and Maintenance: A Practitioner's Approach, John Wiley & Sons, ISBN:0470603410, 9780470603413
- Pigoski, Thomas, 1997 Practical Software Maintenance, M. New York, Wiley Computer Publishing, ISBN 0471170011
- Lewis, William, 2000 Software Testing and Continuous Quality Improvement,. CRC Press. ISBN 0849398339

1.7.6 BSE3106 Mobile Networks and Computing

Course Level	: Year III Semester I
Course Credit	: 3
Contact Hours	: 45

i. Course Description

This course examines principles, design, implementation, and performance of mobile computing and wireless networking. The aim is to lay a foundation in the student's understanding and skills in mobile computing and wireless networking standards, technologies, application and services. Ideally the course is an integration of Wireless Networking and Mobile Computing.

ii. Course Objectives

The objectives of this course are, to;

- Introduce the theory and practice of Mobile networking and computing.
- Facilitate the development of technical skills in mobile application development platforms particularly JME and Android.
- Empower students skills in application of Java and other technologies to mobile application development
- Build the thinking around concepts in wireless technology and mobile computing including standards, technologies, devices and services
- Build handson skills in new technology and cutting-edge projects

iii. Learning outcomes

At the end of this course, the student should;

- Be in position to demonstrate knowledge of mobile computing and wireless networking.
- Be able to discuss the considerations in wireless mobile networking architectures.
- Confortably design and implement wireless and mobile networks using Bluetooth, Wi-Fi among others
- Be able to work with the JME and or Android platforms with minimal difficulty
- Be in position to develop simple mobile applications deployable on Java and/or Android enabled mobile devices

iv. Mode of delivery

In this course a combination of lectures and practical projects will be used. The course is to be taught with ODeL provision

v. Indicative content

- Introduction to Mobile Networks and Computing (6 Hours)
  Applications Involving Wireless Communication
  Effects of Mobility of Devices
  Issues in Cellular Mobile Networks
  Issues in Ad Hoc Networks
  Issues in Cognitive Radio Networks
  Issues in Sensor Networks
- Wireless Network technologies - GSM/GPRS/3/4/5/6G & Wireless LANs)
  (9 Hours)
- Convergence networks, NextGen, (6 Hours)
- Mobile IP, (6 Hours)
- wireless ATM, (6 Hours)
- Wireless Ad Hoc Networks, Bluetooth and other protocols. (6 Hours)
- location–based services (iSMS/MMS/SIM, WAP, I-mode & JME, E)
  (6 Hours)

vi. Assessment Method

- Course work (Tests, group assignments (take home assignments, case studies, individual / group projects)) 30%
- Final written exam: 70%

vii. Reading List

- Koushik Sinha, Sasthi C. Ghosh, Bhabani P. Sinha, 2016, Wireless Networks and Mobile Computing, Ed 1, ISBN 9781482227932, Chapman and Hall/CRC
- Krishna Kant Singh, Akansha Singh, Korhan Cengiz, Dac-Nhuong Le, 2020, Machine Learning and Cognitive Computing for Mobile Communications and Wireless Networks, Edition 1, Wiley-Scrivener, ISBN-10: 1119640369
- Yulei Wu, Sukhdeep Singh, Tarik Taleb, Abhishek Roy, Harpreet S. Dhillon, Madhan Raj Kanagarathinam, Aloknath De, 2021, 6G Mobile Wireless Networks (Computer Communications and Networks) 1st ed. 2021 Edition, ISBN-13: 978-3030727765, Springer.
- Frank H.P. Fitzek, Fabrizio Granelli, Patrick Seeling, 2020, Computing in Communication Networks: From Theory to Practice 1st Ed, ,ISBN-10 : 0128204885 , Academic Press; 1st edition
- Henry Zárate Ceballos, Jorge Parra Amaris, Hernan Jiménez Jiménez, Diego Romero Rincón, Oscar Agudelo Rojas, Jorge Eduardo Ortiz Triviño, 2021, Wireless Network Simulation: A Guide using Ad Hoc Networks and the ns-3 Simulator, ISBN-10 : 1484268482, Apress; 1st ed.
- Jim Doherty, 2015, Wireless and Mobile Device Security: Print Bundle (Jones & Barlett Learning Information Systems Security & Assurance) Illustrated Edition, ISBN-10 : 1284059278 , Jones & Bartlett Learning
- Paul Goransson, Chuck Black, Timothy Culver, 2016, Software Defined Networks: A Comprehensive Approach 2nd Edition, Morgan Kaufmann; ISBN-10 : 0128045558

1.8 Year III Semester II

1.8.1 BSE3210 Software Architecture and Patterns

Course Level: Year III Semester II
Course Credit : 3
Contact Hours : 45

i. Course Description

Based on the fact that most software is build on existing libraries, frameworks, and components, this course teaches the principles and concepts involved in the analysis and design of large software systems. The key software architecture building blocks must be carefully integrated to ensure that the resulting applications are robust and maintainable. This course will address issues of large-scale software development usually referred to as “software architecture”, including architectural design and documentation, component models and technologies, software product lines, frameworks, and design patterns.

ii. Course Objectives

The objectives of this course are, to;

- Enable the student appreciate the role of software architectures and patterns in large scale software systems
- Demonstrate to the student how to design software architectures and patterns for large scale systems in organizations and identify and assess the quality attributes of a system at the architectural level.

iii. Learning outcomes

After successfully completing this course you will be able to;

- Argue the role of software architectures and patterns in large scale software systems.
- Demonstrate how to design software architectures and patterns for large software systems in organizations and identify and assess the quality attributes of a system at the architecture level.

iv. Mode of delivery

Teaching and learning will be in form of classroom lectures, demonstrations student's practical projects, class discussions and quiz

The course is to be taught with ODeL provision

v. Indicative content

1. Introduction to the fundamentals of software architecture. (8 Hours)

2. Software architecture and quality requirements of a software system. (9 Hours)

3. Fundamental principles and guidelines for software architecture design, architectural styles, patterns and frameworks. (8 Hours)

4. Methods, techniques and tools for describing software architecture and documenting design rationale. (9 Hours)

5. Software architecture design and evaluation processes. (10 Hours)

vi. Assessment Method

- Course work (Tests, group assignments (take home assignments, case studies, individual / group projects)) 30%
- Final written exam: 70%

vii. Reading List

- Mark Richards, 2015 Software Architecture Patterns, Publisher(s): O'Reilly Media, Inc. ISBN: 9781491924242
- Ben Stopford, 2018, Designing Event-Driven Systems, Publisher(s): O'Reilly Media, Inc. ISBN: 9781492038245
- Vladik Khononov 2019, What Is Domain-Driven Design? Publisher(s): O'Reilly Media, Inc. ISBN: 9781492057796
- Neal Ford, Rebecca Parsons, Patrick Kua, 2017, Building Evolutionary Architectures, Publisher(s): O'Reilly Media, Inc. ISBN: 9781491986363
- Taylor, R., Medvidovic, N., Dashofy, E.,2010, Software Architecture: Foundations, Theory, and Practice, Wiley.
- Clements, P., Bachmann, F., Bass, L., Garlan, D., Ivers, j., Little, R., Nord, R. and Stafford, J., 2002, Documenting Software Architectures: Views and Beyond, Addison-Wesley.
- Clements, P., Kazman, R. and Klein, M., 2002, Evaluating Software Architectures: Methods and Case Studies, Addison-Wesley.
- Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., Stal, M., 1996, Pattern-Oriented Software Architecture: A System of Patterns, 1996, Wiley.

1.8.2 BSE3211 Software Testing and Verification

Course Level: Year III Semester II
Course Credit : 3
Contact Hours : 45

i. Course Description

The course focuses on testing tools, techniques and methods that can be used to assess the quality and correctness of software systems. The course brings understanding on how these methods, techniques and tools can be used in a software development project. Students will get hands-on experience in building a model for testing and they use this model for both testing and verification purposes. The course is provided in the so-called mini-modules format, which combines lectures and supervised practical work with exercises in small groups.

ii. Course Objectives

The objectives of this course are, to;

- Introduce to the student the various testing tools, techniques and methods that can be used to assess the quality and correctness of software systems.
- Enable the student understand how these methods, techniques and tools can be used in a software development project to increase the software quality.

iii. Learning Outcomes

After completion of the course the student is expected to be able to:

- Knowledge **and understanding**
  - describe the distinction between software verification and software validation
  - name and describe the basic concepts on testing, as well as different testing techniques and approaches
  - describe the connection between software development phases and kinds of testing
  - exemplify and describe a number of different test methods, and be able to use them in practical situations
- **Skills and abilities**
  - write models in at least one formal specification language
  - construct appropriate and meaningful test cases, and interpret and explain (to stakeholders) the results of the application of such test cases (using appropriate tools) to practical examples
  - plan and produce appropriate documentation for testing
  - apply different testing techniques on realistic examples
- **Judgement and approach**
  - exemplify and describe tools used for testing software, and be able to use them and interpret their output
  - exemplify and describe the area of formal verification in general, including model checking and runtime verification, and its relationship to software quality
  - identify and hypothesize about sources of program failures, and reflect on how to better verify the correctness of such programs

iv. Indicative Content

- Testing tools, techniques and methods that can be used to assess the quality and correctness of software systems (15 Hours)
- Application of testing techniques and tools in software development project (15 Hours)
- Overview of other verification techniques (15 Hours)

v. Mode of delivery

Relevant textbooks, Testing software like Appium, Junit, Maven, Selenium, power point slides, a projector, laptop and necessary teaching aids. Lectures and practical group work with exercises will be used.

The course is to be taught with ODeL provision

vi. Mode of Assessment

• Mini-project assignments and one test	30%
• Final written examination	70%

vii. References

- Paul Felten, 2017, Software Testing Basics: Software Verification Fundamentals for Dedicated Testers in the Medical Device Industry 1st Edition, ISBN-10 : 1543269540
- Mukesh Sharma 2016, Software Testing 2020: Preparing for New Roles 1st Edition, Auerbach Publications; 1st edition, ISBN-10 : 9781498788878
- Beizer, Boris. 2011, Software testing techniques. Dreamtech Press, **2003.**
- Khannur, Arunkumar. Software Testing: Techniques and Applications. Pearson Education India.
- Dasso, Aristides, 2006, Verification, validation and testing in software engineering. IGI Global.

1.8.3 IST2203 Research Methodology

Course Level : Year III Semester II
Course Credit : 4
Contact Hours : 60

i. 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.

ii. Course objectives

The 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 collect and analyze data
- Document and present findings

iii. Learning outcomes

After successfully completing this course you 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.

iv. Mode of delivery

Teaching and learning will be in form of classroom lectures, demonstrations, research scholarly material analysis, class discussions and quiz.

The course is to be taught with ODeL provision

v. Indicative content

1. 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. (7 Hours)
2. Research methods: Student learns the various quantitative and qualitative research methods. Student also learns the data gathering techniques and statistical data analysis tools. (8 Hours)
3. 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? (15 Hours)
4. Research application: covers the practical application of concepts learned from parts 1 to 3. The lecturer suggests one or several class projects that require practical application of concepts learned from parts 1 to 3. (15 Hours)
5. 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 etc. (15 Hours)

vi. Assessment Method

- Course work (Tests, group assignments (take home assignments, case studies, individual / group projects)) 30%
- Final written exam: 70%

vii. Reading List

- Zikmund, W.G. (2010). Business Research Methods (8th Ed.) Thomson South Western
- Leedy, P.D., & Ormrod, J.E. (2004). Practical Research: Planning and Design Paperback, Prentice Hall
- Sounders, M., Lewis, P. & Thornhill, A. (2003). Research Methods for Students. 3rd edition, UK, Financial times, Prentice hall
- Graziano, A.M, Michael, L. (2006). Research Methods: A process of Inquiry Raulin, Hardcover, Prentice Hall

1.8.4 CSC2206 Machine Learning

Course Level : Year III Semester II

Course Credit : 4
Contact Hours : 60

i. 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 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 the main models and algorithms for regression, classification, clustering and Markov decision processes. Topics will include: machine learning basics, classification model (KNN, Naive, decision tree and so on), linear and logistic regression, regularization, probabilistic (Bayesian) inference, support vector machines and kernel methods, artificial neural network/ convulsions neural networks and its variants, deep learning, clustering, and dimensionality reduction. Students taking the course will need to be familiar with linear algebra, probability theory, and programming specifically in Python.

ii. Course Objectives

The obejctives of this course are;

- To introduce methods for learning from data, and provide the necessary mathematical background
- To impart knowledge of how the methods work, how to evaluate the performance a machine learning system and how to get the best performance from them.

iii. 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 machine learning algorithms.
- understand how to apply a variety of machine learning algorithms to data.
- understand how to perform evaluation of machine learning algorithms and model selection.

iv. Indicative Content

Topic 1: Introduction to machine learning (6 hours)

This topic introduces the students to the major concepts of machine learning. It introduces the broad concepts of learning. Concepts governing Supervised Learning, Unsupervised Learning, Data Representation, and Overfitting will be discussed in this topic.

Topic 2: Machine Learning Basics (10 hours)

The topic will introduce the basis concepts and distributions used in building machine learning models. In the topic we discuss the Bayesian Theorem, Gaussian Distribution, Gaussian Mixture Models and Maximum Likelihood

Estimation technique.

Topic 3: Supervised Learning Models (20 hours)

The topic introduces various supervised learning models in detail. In the topic we discuss Linear Regression, Logistic Regression, Naive Bayes Classifier , Decision Tree, Support Vector Machines(SVM), kernel SVM and K-nearest Neighbours Classifier. We also discuss Regularisation and Gradient Descent.

Topic 4: Unsupervised Learning (8 hours)

This topic introduces the idea of learning from data by clustering. We discus the K-Means and Expectation Maximisation methods.

Topic 5: Introduction to Deep Learning (8 hours)

The topic provides an introduction to Deep learning. Concepts concerning Multilayer Perceptron, Convolutional Neural Network and Recurrent Neural Network will be discussed.

Topic 6: Feature selection and Evaluation metrics (8hours)

In the last Topic we discuss methods of feature selections and the various methods for evaluating the models.

v. Mode of delivery

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. The course is to be taught with ODeL provision.

Teaching and learning will be implemented through 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.

vi. Mode of Assessment

Assessment will be by continuous assessment through programming assignments (20%) , written assignments and quizzes (20%) and Final exam(practical (30%) and written (30%))

vii. References

1. Bill Hanson, Kevin Tromp, 2019, Machine Learning 2020: The Ultimate Guide to Data Science, Artificial Intelligence, and Neural Networks in Modern Business and Marketing applications: The Data Science Guide, Bill Hanson
2. Bishop, Christopher, 2006 Pattern Recognition and Machine Learning,Springer-Verlag New York, ISBN 978-0-387-31073-2.
3. Tom Mitchell. Machine Learning. McGraw Hill.
4. Andreas Mueller. Scikit-Learn Tutorial: Statistical-Learning for Scientific Data Processing.
5. Shai Shalev-Shwartz and Shai Ben-David Understanding Machine Learning: From Theory to Algorithms.

6. Stephen Marsland. Machine Learning: An Algorithmic Perspective.
7. Willi Richert and Luis Pedro Coelho. Building Machine Learning Systems with Python

1.8.5 BSE3214 Cloud Computing and Big Data

Course Level : Year III Semester II
Course Credit : 4
Contact Hours : 60

i. Course overview

Cloud Computing has transformed the IT industry by opening the possibility for infinite or at least highly elastic scalability in the delivery of enterprise applications and software as a service (SaaS). Amazon Elastic Cloud, Microsoft's Azure, Google App Engine, and many other Cloud offerings give mature software vendors and new start-ups the option to deploy their applications to systems of infinite computational power with practically no initial capital investment and with modest operating costs proportional to the actual use. This course serves as an introduction to cloud computing for individuals and organizations. With the recent growth in cloud computing, the real value of cloud computing is not just in providing cloud servers but rather in the data produced by many of the cloud services. Big data is one of the most commonly used terms in the world of cloud computing. This course examines some of the most common technologies used in big data today.

ii. Courser objectives

The objective of this course is to;

- Provide students with the comprehensive and in-depth knowledge of Cloud Computing concepts, technologies, architecture and applications
- Introduce research on state-of-the-art in Cloud Computing fundamental issues, technologies, applications and implementations.
- Expose students to big data that comes along with cloud computing and ways to store, manage & analyse big datasets for decision-making.

iii. Learning outcomes

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

- Articulate the main concepts, key technologies, strengths, and limitations of cloud computing and the possible applications for state-of-the-art cloud computing.
- Identify the architecture and infrastructure of cloud computing, including SaaS, PaaS, IaaS, public cloud, private cloud, hybrid cloud, etc.
- 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.
- Analyze the performance, scalability, and availability of the underlying cloud technologies and software.
- Perform an economic and benefits analysis of cloud computing business models
- Explain the core issues of cloud computing such as security, privacy, and interoperability.
- Demonstrate how to store, manage, analyze unstructured data and process

big data.

- Demonstrate how big data analytics affect society

iv. **Indicative Content**

Part 1: Overview of Cloud Computing (10 Hours)

- Advantages, History, and Characteristics of Cloud Computing
- Service & Deployment Models, Infrastructure, and Consumer View
- Security and Scenarios
- Assumptions, Terms of Service, & Promises
- Limitations, Obligations, Recommendations, & Implications

Part 2: Dynamic Interactions and Computing Architectures (15 Hours)

- Service, Deployment, Scope, and Control
- SaaS Interaction Dynamics and Software Stack Control
- SaaS Benefits, Issues and Concerns, Suitability, and Recommendations
- PaaS Dynamics and Software Stack Control
- PaaS Benefits, Issues and Concerns, Suitability, and Recommendations
- IaaS Abstract Interaction Dynamics and Software Stack Control
- IaaS Operational View
- IaaS Benefits
- IaaS Issues and Concerns, and Recommendations

Part 3: Economics of Cloud Computing (15 Hours)

- Review of Service Models
- SWOT Analysis and Value Proposition
- General Cloud Computing Risks
- Cloud Computing Cost Analysis
- Selecting an IaaS Provider
- Cloud Standards and Inter cloud Interoperability
- Recommendations for Successful Cloud Migration

Part 4: Big Data & Analytics (20 Hours)

- An Introduction to Data Analytics
    - Structured Versus Unstructured Data
    - Data in Motion Versus Data at Rest
    - Data Analytics Challenges
- Machine Learning Overview
    - Supervised Learning
    - Unsupervised Learning
    - Machine Learning and Getting Intelligence from Big Data
    - Predictive Analytics
- Big Data Analytics Tools and Technology
    - Massively Parallel Processing Databases
    - NoSQL Databases
    - Hadoop
    - YARN
    - The Hadoop Ecosystem
    - Apache Kafka

    - Lambda Architecture
- Edge Streaming Analytics
    - Comparing Big Data and Edge Analytics
    - Edge Analytics Core Functions
    - Distributed Analytics Systems
- Network Analytics
    - Flexible NetFlow Architecture

v. Mode of delivery

- Lectures, seminars & laboratory exercises
- The course is to be taught with ODeL provision

vi. Assessment

- Test, Projects, presentations (30%)
- Final Examination (70%)

vii. Reading References

- Cloud Computing: Concepts, Technology & Architecture 2013, By Richardo Puttini, Thomas Erl, and Zaigham Mahmood, 2013
- Ray Rafaels, 2018, Cloud Computing, 2nd Ed., CreateSpace Independent Publishing Platform, ISBN-10 : 1986726282
- Kevin L. Jackson and Scott Goessling , 2018, "Architecting Cloud Computing Solutions: Build cloud strategies that align technology and economics while effectively managing risk" Packt Publishing, ISBN-13 : 978-1788472425
- IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things: 2017, by David Hanes
- Other Materials: Several modules will be based on recent conference/journal papers as well as documentation from cloud providers

1.8.6 BSE3213 Embedded Systems II

Course Level	: Year III Semester II
Course Credit	: 4
Contact Hours	: 60

i. Course Description

The course covers advanced concepts of embedded systems. It shall start with theoretical knowledge on real-time system concepts. Later, the embedded operating systems shall be covered, while blending theory with practicals. Protocols for embedded systems , emerging technologies covered, security and power aware embedded computing shall be introduced and demonstrated. All the theory shall be summed up in an implementation of a robotics project.

ii. Course Objectives

The main objective of this course is to advance the concepts covered in the introductory course of Embedded Systems. The focus is on the applicability of the concepts covered in this introductory course.

iii. Learning Outcomes

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

- Appreciate the Intenet of Things technologies and their applications
- Demonstrate practical skills in the design of embedded systems in different applications.

iv. Indicative content

1. Internet of Things (IoT) technologies and applications	(6 Hours)
2. Real-time systems	(7 Hours)
3. Embedded operating system	(7 Hours)
4. Protocols for embedded systems	(6 Hours)
5. Emerging technologies/applications/principles	(6 Hours)
6. Power-Aware Embedded Computing	(6 Hours)
7. Embedded systems security	(7 Hours)
8. Robotic systems project	(15 Hours)

v. Learning Outcomes

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

- Demonstrate comprehensive knowledge in understading and programming embedded systems operating systems
- Construct a robotic prototype and application design based on an embedded operating system
- Understand the applications of Internet Of Things
- Understand and practically implement protocols for embedded systems

vi. Pre-requisite: BSE3111 Embedded Systems I

vii. Mode of delivery

- Using formal face-to-face lectures, to provide a focus on the core analytical material in the course, together with qualitative, alternative explanations to aid your understanding.
- Practical demonstrations to support the formal lecture material and also provide you with practical construction, measurement and debugging skills
- One take-home project to assess the understanding of the concepts
- The course is to be taught with ODeL provision

viii. Mode of Assessment

Coursework in form of tests and individual/group projects will contribute 30% and the final exam will contribute 70%.

ix. Reference Material

1. Dey, Nilanjan, and Amartya Mukherjee. Embedded systems and robotics with open source tools. CRC Press, 2018.
2. Wang, Jiacun. Real-time embedded systems. John Wiley & Sons, 2017.
3. Zurawski, Richard. Embedded Systems Handbook: Embedded systems design and verification. CRC press, 2018.
4. Alioto, Massimo, ed. Enabling the Internet of Things: From Integrated Circuits to Integrated Systems. Springer, 2017.

1.9 Year III Recess

6.9.1 BSE3302 Field Attachment

Course Level	: Year III Recess
Course Credit	: 5
Contact Hours	: 75

While this course is indicated at being conducted during the Year 3 recess term, students who can find field attachment placement during the Year 3 end Semester I university break (December to January) and can carry on with Field attachment into the first few weeks Semester 2, particularly the evening students will be allowed to undertake their internship starting at the end of Year 3 Semester

I exams.

i. **Course Description**

This course assumes the students have gained a good amount of practical and professional skills to practice in the field i.e. having successfully completed BSE 1301 & BSE 2301. This course provides students with a mentorship opportunity in a real work environment

ii. **Course Objectives**

The objectives of this course are, to:

- equip students with practical skills in software engineering from requirements solicitation to testing;
- Mentorship students in writing professional software products;
- Provide students an opportunity to build team-work, communication, appearance and leadership skills needed in the workforce

iii. **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.

iv. **Indicative Content**

The students shall acquire hands on training emphasing the software development process. Other relevant areas include e.g. maintenance, servicing and troubleshooting, and training in computer skills and applications as well as software development. Hardware maintenance; Practicals; On Job training; Report writing.

v. **Mode of delivery**

Mentorship, Practicals, & group work

The course is to be taught with ODeL provision

vi. **Mode of Assessment**

Student Journal, Field Supervisor form, Academic supervisor's form and field attachment report

1.10 Year IV Semester I

1.10.1 BSE4100 Software Engineering Project I

Course Level : Year IV Semester I
Course Credit : 5
Contact Hours : 75

i. **Course Description**

The course covers practical aspects of the initial phases of a software engineering. It ensures practical application of skills in requirements gathering, and design and specification

ii. **Course Objectives**

The objectives of this course are, to;

- Enable the student gain independent skills in collecting requirements, documenting user requirements and technical requirements for non-trivial software engineering/research projects.

- Impart skills to practically specify, design and implement a software project, while working with an academic supervisor.

iii. Course Learning Outcomes

Upon successful completion of this course the student will have ability to:

- Demonstrate independent skills in collecting requirements, documenting user requirements and technical requirements for non-trivial software engineering/ research projects by pursuing a lengthy Software engineering project.
- Demonstrate skills of specifying, designing and implementing a project, with assistance of one of the Professors/ Lecturers as adviser/ supervisor.

iv. Mode of delivery

Lecture notes, student presentations, short exploratory assignments

The course is to be taught with ODeL provision

v. Indicative Content

The student develops a framework within which research will be conducted and offers evidence of qualifications to pursue the research. Concepts and theories underlying the student's Project research are articulated, the problem is clearly stated, and specific, measurable goals are specified, a literature review is presented, the methods of conducting research are delineated, and strategy to achieve the goal is given.

NB: The time allocated to tasks to be done in this course will depend on the nature of the project being done by the students and the schedules arranged between the students and the their supervisors. However, the maximum total contact hours for the course is 75.

vi. Assessment

- Course work (Tests, group assignments (take home assignments, case studies, individual / group projects)) 30%
- Final written exam: 70%

1.10.2 BSE4102 ICT Innovation and Entrepreneurship

Course Level : Year IV Semester I
Course Credit : 3
Contact Hours : 45

i. Course Description

ICT innovation is accelerating Africa's growth by finding better ways of solving local problems, even as it attracts top technology global brands. Young Africans are unleashing entrepreneurial energies as governments continue to enact reforms that improve business environments. An increasing number of start-ups are providing solutions to different business problems in the region. Despite this progress, Africa is still deeply underperforming in core areas that will redesign its economy and make it more sustainable. Information and Communications Technologies (ICT) is driving efficiency in business operations across Africa from mobile payment to integrated banking systems.

ii. Learning Objectives

The objectives of this course will include;

- Exploration of the underpinnings and prospects for the future of ICT and its applications in sub-Saharan Africa. Focusing on the most recent literature and reports on economic and technology development in Africa,
- To understand the critical factors that will determine how technology will develop in Africa and what business models are likely to succeed in creating economically viable ICT enterprises in this rapidly changing part of the world.
- To explore different types of entrepreneurial enterprise and innovation activity in Africa's innovation system, and how these activities exist in a global context

iii. Learning Outcomes

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

- Describe the role of innovation and entrepreneurship in African society.
- Explain and illustrate theories of business innovation and entrepreneurship, the evolution of industries and economies, and the roles of entrepreneurs.
- Develop a comprehensive and well structured business plan for a new venture.
- Present a persuasive business plan to potential investors or to internal stakeholders and effectively answer probing questions on the substance of the plan.
- Work effectively in multidisciplinary, cross-cultural teams, communicating, negotiating and contributing shared contributions towards the development of a team project.
- Make start-ups and spinoffs.

iv. Indicative Content

1.	ICT idea generation and entrepreneurial thinking	(6 Hours)
2.	ICT Innovation management	(6 Hours)
3.	Opportunity spotting and evaluation	(6 Hours)
4.	The Power of Story to fuel innovation	(6 Hours)
5.	Industry and market research	(6 Hours)
6.	Financial forecasting and business plans	(6 Hours)
7.	Pitching for resource providers and negotiating deals	(6 Hours)
8.	Licensing and patenting	(3 Hours)

v. Pre-requisite: None

vi. Mode of delivery

Live lectures, inclass group presentations, case study reviews and group discussions.

The course is to be taught with ODeL provision

vii. Assessment

Assignments on business interviews and pitching, business projects, Creativity and innovation (30%), Final Examination (70%)

viii. Reference List

- John R. Bessant and Joe Tidd, 2015, Innovation and Entrepreneurship, 3rd Edition, ISBN: 978-1-118-99309-5
- Edward I. Huizenga, 2005, Innovation Management in the ICT Sector: How Frontrunners Stay Ahead, Edward Elgar Publishing Ltd ISBN-10 : 1845422244, ISBN-13 : 978-1845422240
- Tobias Kollmann and Andreas Kuckertz, 2010, E-entrepreneurship and ICT ventures strategy, organization and technology, ISBN-13 : 978-1616922498, ISBN-10 : 1616922494
- Thomas N. Duening, Robert A. Hisrich, Michael A. Lechter, 2014, Technology Entrepreneurship: Taking Innovation to the Marketplace 2nd Edition, Academic Press, ISBN-10 : 012420175X, ISBN-13 : 978-0124201750

1.10.3 BSE4104 Emerging Trends in Software Engineering

Course Level : Year IV Semester I
Course Credit : 3
Contact Hours : 45

i. Course Description

Technology, specifically software industry is growing very fast in this 21st century more than ever before. New software innovations come on the market and are quickly overwritten by new ones, either due to competition from the software industry or due to demand for new tsoftware solutiions from the consumers. A software engineering student needs to be aware of the latest trends in software industry and adjust on his/her skills to remain relevant to software industry. This course looks at the latest software tech-innovations, tools, frameworks, standards among others and how they are applied in software industry.

ii. Course Objectives

The objectives of this course are, to;

- Equipe the student with the knowledge about the latest tech-innovations, tools, frameworks, standards among others in the software industry.
- Enable the student apply the latest solutions in software industry into solving software engineering tasks.

iii. Learning outcomes

At the end of this course, the student should;

- Demonstrate knowledge about the latest software tech-innovations, tools, frameworks, standards among others in the software.
- Demonstrate practical skills in applying the latest software solutions in software industry to solve particular software engineering tasks.

iv. Mode of delivery

Review/analysis of trending scholarly work in software engineering, presentations and discussion. The course is to be taught with ODeL provision

v. Indicative content

No particular indicative content but the course instructor should identify areas of discussion and focus should be on studying and analyzing the state-of-the-art and state-of-practice literature.

vi. Assessment Method

Courseworks in form of tests, individual and groupwork take home assignments will contribute 30%, final exam will contribute 70%.

vii. References

No particular references for this course. However, the course instructor will identify the latest literature about the emerging trends in the field of software engineering.

1.10.4 BSE4105 Software Integration and Deployment

Course Level : Year IV Semester I
Course Credit : 4
Contact Hours : 60

i. Course Description

The course unit aims to address current industry standards in development, delivery and deployment of software applications coupled with best practices. Building and delivering software isn't easy. There is need for consistent quality and ability to deliver new functionality quickly. This course will introduce practices of Continuous Integration and Continuous Delivery for building and delivering quality software in a reliable way.

ii. Course Objectives

The objectives of this course are, to;

1. Explore the different version control systems
2. Understand the industry software development techniques
3. Aquaint students with the continuous integration/delivery/deployment tools

iii. Learning outcomes

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

- Demonstrate competence in using different version control sysems
- Articulate and apply various software development techniques.
- Use different continuous software integration/delivery/deployment tools on software development projects.

iv. Mode of delivery

Teaching will be in terms of lectures, class presentations and practical demonstrations. A working project through the course's delivery will be administered. The course is to be taught with ODeL provision.

v. Indicative content

1. Version Control systems such as ClearCase, CVS, Git, Mercurial, Perforce, StarTeam, Subversion, Team Foundation Server, SourceGear Vault, and Visual SourceSafe for Continuous Integration (15 Hours)
2. Activity & Task management tools such as Trello (15 Hours)
3. Build and Test automation (10 Hours)
4. Continuous Integration using CI tools such as Jenkins, GitLab, Travis CI, Bamboo, TeamCity, Spinnaker and CircleCI (10 Hours)
5. Software development and issues tracking e.g JIRA (10 Hours)

vi. **Assessment Method**

- Course work (Tests, group assignments (take home assignments, case studies, individual / group projects)) 30%
- Final written exam: 70%

vii. **References**

- John Ferguson Smart, 2011, Jenkins: The Definitive Guide: Continuous Integration for the Masses, 1st ed. , O'Reilly Media, ISBN-10 : 9781449305352
- Sander Rossel, 2017, Continuous Integration, Delivery, and Deployment Reliable and faster software releases with automating builds, tests, and deployment 1 ed., Packt Publishing, ISBN: 978-1491918236

1.11 Year IV Semester II

1.11.1 BSE4200 Software Engineering Project II

Course Level : Year IV Semester II
Course Credit : 5
Contact Hours : 75

i. **Course Description**

The course covers practical aspects of the later phases of a software engineering. It ensures practical application of skills in implementation, testing and deployment

ii. **Course Objectives**

Upon successful completion of this course the student will have ability to:

- Demonstrate independent skills in implementing non-trivial software engineering/ research projects by pursuing a lengthy Software engineering project.
- Demonstrate skills of Documenting, deploying a testing a well engineered solution, with assistance of one of the Professors/ Lecturers as adviser/ supervisor.

iii. **Course Content**

The student implements, documents, tests and deploys a software solution using the state of the art principles, concepts and technologies

The **specific deliverables** are:

- Software Implementation using state of the art technologies
- Detailed software documentation in accordance to well known practices
- Installation Manuals
- User Manual
- Testing and validation strategy.

iv. **Pre-requisite:** BSE4100 Software Engineering Project I

v. **Assessment Method**

Project report and Oral presentation (100%)

vi. **References**

No particular reference will be used for this course unit, except whatever is recommended by the student supervisors.

1.11.2 BSE4202 Software Security

Course Level : Year IV Semester II

Course Credit : 4
Contact Hours : 60

i. **Course Description**

The course unit addresses the common security problems in software as well as their underlying causes. It then addresses the techniques, guidelines, principles and tools that prevent or detect them.

ii. **Course Objectives**

The objectives of this course are, to;

- Explore the common security problems in software
- Learn the main causes of the security problems in software
- Teach the techniques of detecting security flaws in software during implementation

iii. **Learning outcomes**

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

- Explain and and solve common security probllems in software
- Identify the main causes of security problems in software and how they should be avoided.
- Apply the techniques of detecting security flaws in software during implementation.

iv. **Mode of delivery**

Teaching will be in terms of lectures, class presentations and practical demonstrations. The course is to be taught with ODeL provision

v. **Pre-requisite:** None

vi. **Indicative content**

1. Usability and Psychology: Introduction to Phishing and Social Engineering, and Countermeasures; Password Issues: Password Entry, Remembering the Password, Naive Password Choice, Password Manglers, Password attacks and controls. (8 Hours)
2. Security Protocols and Attacks: Managing Encryption Keys, Basic Key Management, The Needham-Schroeder Protocol, Kerberos, Practical Key Management. (8 Hours)
3. Access Control: Operating System Access Controls, Groups and Roles, Access Control Lists, Unix Operating System Security, Windows OS Security; Advanced Protection Techniques: Sandboxing and Proof-Carrying Code, Virtualization, Trusted Computing. (8 Hours)
4. Cryptography: Stream Ciphers, Block Ciphers and Hash functions; Public Key Cryptography. (10 Hours)
5. Multilevel and Multilateral security: Bell-LaPadula model, Biba Model, Data flow models (Lattice and Chinese wall), Inference Control. (8 Hours)
6. API Security: API risks and attacks; Full screen API attacks, Cross Site scripting, SQL injections and buffer overflow attacks. (8 Hours)
7. Network attacks and defences: Network protocols, malware, configuration management, intrusion detection. (10 Hours)

vii. **Assessment Method**

- Course work (Tests, group assignments (take home assignments, case studies, individual / group projects)) 30%
- Final written exam: 70%

viii. **References**

- Gerardus Blokdyk, 2019, Security Engineering Requirements A Complete Guide, 5STARCooks, ISBN-10 : 0655916962
- Gerardus Blokdyk 2019, Security Engineering A Complete Guide – Practical tools for self assessment, Emereo Publishing , ISBN-13:9780655977681
- Michael Howard, David LeBlanc, John Viega , 2009, 24 Deadly Sins of Software Security: Programming Flaws and How to Fix Them, 1st ed, McGraw-Hill Education, ISBN-10 : 0071626751
- Lee Brotherston, Amanda Berlin, 2017, "Defensive Security Handbook" O'Reilly Media, Inc., ISBN: 9781491960387

1.11.3 BSE4203 Software Engineering Standards and Ethics

Course Level : Year IV Semester II
Course Credit : 3
Contact Hours : 45

i. Course Description

Software engineering ethics can be approached from three directions. First, it can describe the activity of software engineers making practical choices that affect other people in significant ways. Second, it can be used to describe a collection of principles, guidelines, or ethical imperatives that guide or legislative action, and third, it can be used to name a discipline that studies the relationship between the other two senses of ethics. Software engineering ethics is clearly both an activity and a body of principles. The discipline of software engineering ethics that studies this activity and formalizes these principles, however, is in its infancy. Software Engineering standards define all processes required for developing and maintaining software systems, including the outcomes and/or activities of each process. Standards emphasize conformance to processes and information item content. Standards are therefore a way of ensuring quality in software engineering. Hence, this course develops student understanding about historical, social, ethical and professional issues related to the discipline of computing.

ii. Course Objectives

The objectives of this course are, to;

- Train students on ethical behavior as software engineers and be able to understand the need for personal and professional ethics.
- Develop student analysis skills for ethical implications of software engineering practices and that ethical practice is about not just avoiding harm but doing good.
- Enable the student understand different software standards and how such standards help in software process improvement.

iii. Learning Outcomes

Having successfully completed the course, students should be able to:

- Behave ethically as a software engineer
- demonstrate knowledge and understanding for both personal and professional ethics.
- Explian the ethical implications of software engineering practices that can incur harm.
- Apply five ethically constructive habits of mind and action.

- Apply ethical principles to controversies such as online harassment and privacy.
- Achieve software process improvement through the use of software and systems engineering standards

iv. **Indicative content**

1.	Professional codes of software engineering ethics	(3 Hours)
2.	Harm in software engineering	(3 Hours)
3.	How do software engineers contribute to the good life for others	(3 Hours)
4.	To whom are software engineers obligated	(3 Hours)
5.	Where do ethical obligations come from	(3 Hours)
6.	How can software engineers act ethically	(3 Hours)
7.	Intellectual property	(3 Hours)
8.	Cyber law and security	(3 Hours)
9.	Definition of a software engineering standard	(3 Hours)
10.	Evolution of software engineering standards	(3 Hours)
11.	ISO/IEC 12207:2008 and 2017 standard	(4 Hours)
12.	ISO/IEC 15288 standard	(4 Hours)
13.	Other standards such as ISO/IEC/IEEE 24765, ISO 25000	(4 Hours)
14.	Relationships between software engineering standards	(3 Hours)

v. **Mode of delivery**

The course adopts a student- centered learning involving the following:

- Lecture presentations and guided discovery method
- Classroom presentations
- Case study discussions
- Group projects and learner-assisted discussions on assigned topics
- Learners will be provided with tasks / activities meant to facilitate understanding of concepts of Software Engineering Standards and Ethics
- The course is to be taught with ODeL provision

vi. **Pre-requisite**: None

vii. **Assessment Method**

Practical assignments: 40%; Final examination: 70%.

viii. **Reference List**

- George Reynolds, 2018, Ethics in Information Technology 6th Ed., Cengage Learning; ISBN-13 : 978-1337405874, ASIN : 1337405876
- Michael Quinn, 2016, Ethics for the Information Age, 7th Ed., Pearson, ISBN-13 : 978-0134296548, ASIN : 0134296540
- Schneider, F., & Berenbach, B., 2013. A literature survey on international standards for systems requirements engineering. In Procedia Computer Science (Vol. 16, pp. 796–805). Elsevier B.V. https://doi.org/10.1016/j.procs.2013.01.083

1.11.4 BSE4204 Software Quality Management

Course Level : Year IV Semester II

Course Credit : 3
Contact Hours : 45

i. Course Description

In the past, software quality initiatives often focused on software testing. Now, software quality improvements derive from the design of software development processes - a quality process creates quality products. The course will cover methods and tools for achieving software quality assurance at various levels of a software system including at the module, subsystem, and system levels. The principles of software development and management are presented with special emphasis on the processes and activities of quality assurance. State of the art tools and techniques including development process modeling, manual and computer-assisted reviews, and ROI analysis of new processes. The course will prepare students to methodically develop a software quality-assurance program. This course provides practical knowledge of a variety of quality assurance techniques, and an understanding of some of the tradeoffs between techniques.

ii. Course Objectives

The objectives of this course are, to;

- Teach student the Software Quality Assurance(SQA) context, SQA projects, SQA management and standards .
- Enable the student gain critical thinking skills, integrative reasoning and communication in SQA management.

iii. Learning Outcomes

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

- Explain knowledge of SQA projects, management of SQA on software projects and use of standards to ensure software quality.
- Demonstrate critical thinking skills, integrative reasoning and communication in SQA management.

iv. Indicative content

1. Walkthroughs, Inspections, Reviews, Audits, Checklists, Interviews, Workshops (3 Hours)
2. Traceability analysis (3 Hours)
3. Quality Function Deployment (3 Hours)
4. Failure Mode and Effects Analysis (3 Hours)
5. Risk management (e.g., DDP) (3 Hours)
6. Requirements monitoring (3 Hours)
7. Model Checking (e.g., Spin from ATT) and Model Analysis (e.g., Alloy @ MIT) (3 Hours)
8. Goal-oriented requirements analysis: KAOS, i* (3 Hours)
9. ISO 9000 Quality Assurance (3 Hours)
10. Cost-Value approaches for prioritizing requirements, testing, etc. (3 Hours)
11. Use-case and test-case alignment Software Quality Management Syllabus 6/17/2007 Page 11/15 (3 Hours)
12. Six Sigma methods (e.g., 5 Whys, Fishbone diagrams) and software (3 Hours)
13. Tools: Rational Method Composer, Requisite Pro, Doors, Holosofx (IBM WebSphere Business Modeler), Rational Robot (and other testing tools), etc. (3 Hours)
14. Eclipse Quality Assurance Plugins, especially Jupiter (Document Review), Coverlipse, and the many metric calculation and visualization

tools. (3 Hours)

15. Process simulation (AnyLogic, Holosofx, etc.) (3 Hours)

v. Mode of delivery

The course adopts a student- centered learning involving the following:

- Lecture presentations and guided discovery method
- Role play
- Classroom presentations
- Group projects and learner-assisted discussions on assigned topics
- Learners will be provided with tasks / activities meant to facilitate understanding of concepts of software quality Management

vi. Pre-requisite: None

vii. Assessment Method

Practical assignments: 30%. Final examination: 70%;

viii. Reference List

- Gerard O'Regan, 2014, Introduction to Software Quality - Undergraduate Topics in Computer Science, Springer, ISBN 3319061062, 9783319061061
- Rajiv Chopra, 2018, Software Quality Assurance: A Self-Teaching Introduction, Mercury Learning & Information, ISBN 1683921690, 9781683921691

Appendix A: ACADEMIC STAFF LIST

NAME	Qualifications and Awarding Institution	Area of Specialization		
Networks Department Staff:			**Load**	**status**
Swaib Kyanda Kaawaase, (Ag. HoD)	PhD. Eng (Haerbin Eng. Univ.), MSc. CS (Mak), BSc. Sc(Mak)	Wireless networks, Wireless Net Protocols, Wireless Net security	3	Permanent
Julianne Sansa Otim	PhD. Maths and Nat. Sciences (RUG), MSc. CS (Mak), BSc. (CS, Maths) (Mak)	Internet of Things, Networking Protocols, Network security	6	
Tonny Eddie Bulega	PhD. South China Univ. Of Tech, Guangzhou (SCUTG), MSc. Eng (SCUTG), BSc Eng (Mak)	Mobile Wireless Nets, Wireless Sensor Nets, Communication System	3	Permanent
Johnson Mwebaze	PhD. Maths and Nat. Sciences (RUG), MSc. CS (Mak), PGDCS (Mak), BSc. (CS, Maths) (Mak)	Data Comms and Nets, Scientific Data Processing, Web & Computer Programming	12	Permanent
Benjamin Kanagwa	PhD in Software Engineering, Radboud University	Software Development	9	Permanent
Joseph Balikuddembe	PhD in Computer Science, (UCT)	Software Development	6	Permanent
Drake Patrick Mirembe	PhD in information systems security, University of Groningen	Network Security, Security Protocols	9	Permanent
Moses Ntanda	PhD in Information and computer Science, Xiangtan University	Artificial Intelligencet	6	Permanent
Ruth Mbabazi Mutebi	PhD (Mak), M.Sc (Mak), BSc. Eng (Mak)	Network protocol Design & Development	6	Permanent
Joab Ezra Agaba	PhD (Mak), M.Sc CS (Mak), BSc. (Mak)	Software Development and Reusability in Instructional Design	12	Permanent
Odong Stephen Eyabu	PhD (KNU), M.Sc CS (Mak), BSc. (IUIU)	Sensor Networks	3	Permanent
Kimbugwe Nasser	M.Sc DCSE (Mak), BSc.CS (Mak)	Network	12	Permanent

NAME	HIGHEST DEGREE	Area of Specialization	Load	status
Mary Nsabagwa	PhD (Mak), M.Sc Enginering Hunan Univ, PRC), BSc. (CS) Mak	Embedded systems tech	6	Permanent
Namisanvu Kasawuli Rashida	PhD. (Chalmers UT), M.Sc DCSE (Mak), BSc.Telecom Eng (Kyambogo)	Software	6	Permanent
Bernard Muwonge	M.Sc CS (Mak), BSc. Education (Mbarara Univ. S.T)	Wireless sensors	9	Permanent
Mwotil Alex	M.Sc CS (Mak), BSc. CS (Mak)	Software Defined networks, Data Science	9	Permanent
Grace Kamulegeya	(PhD), M.Sc CS (Uni. Cape Town), BSc. CS(Mak)	Start ups and innovations, big data Science	6	Permanent

Other staff

Computer Science Department Staff:

Engineer Bainomugisha	PhD in Computer Science, (Vrije Universiteit Brussel, Belgium)	Associate Professor	3	Permanent
Barbara Nansamba	M.Sc. DCSE, B.Sc. CS	A.Lecturer	10	Permanent
Joyce Nakatumba-Nabende	PhD. CS (TU Eindhoven), MSc. CS(Mak), BCS (MUST)	Lecturer	3	Permanent

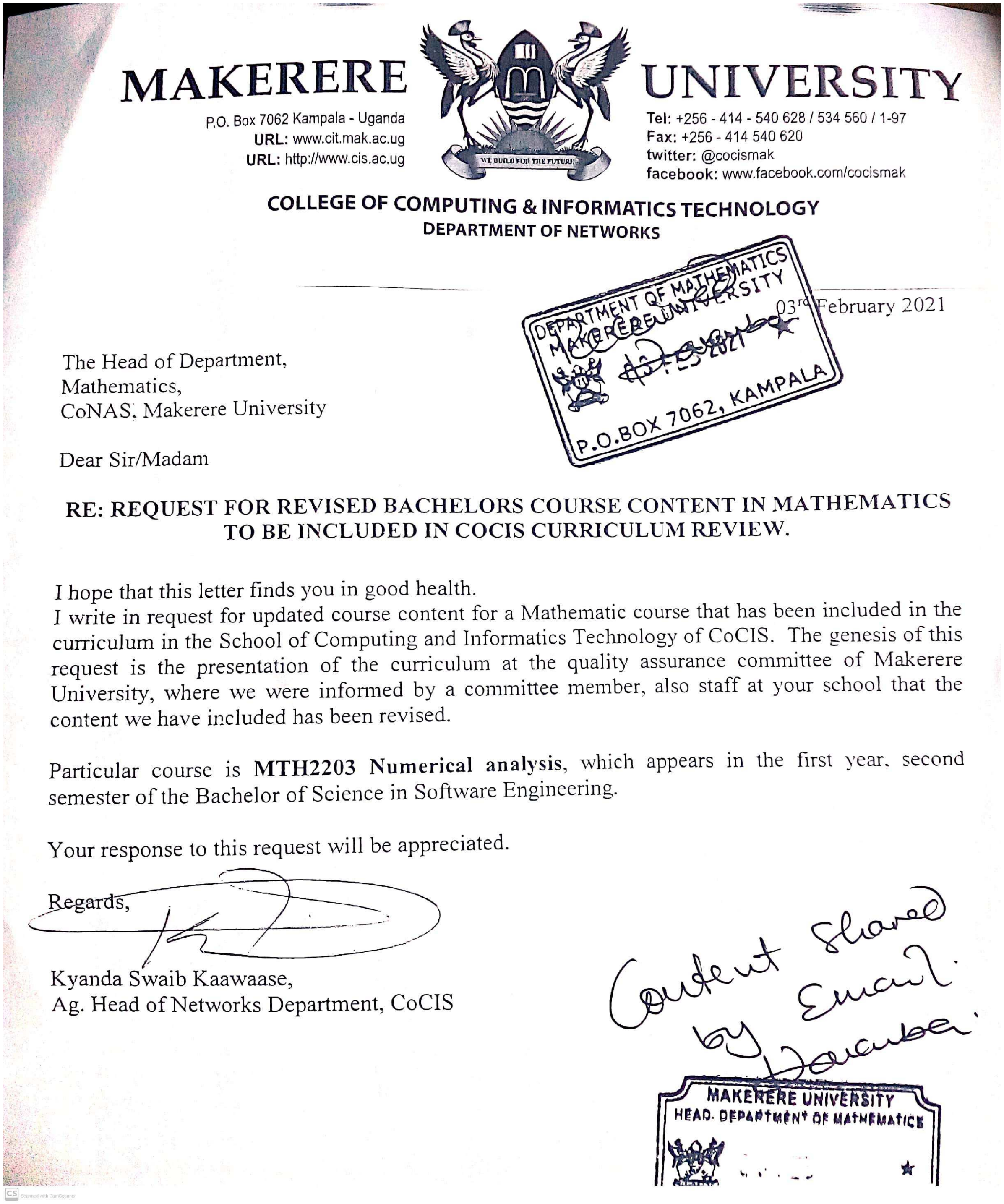
NAME	HIGHEST DEGREE	RANK	Load	status
Information Systems and Information Technology Departments Staff:				
Dr. Rehema Baguma	PhD. IS (RUN2), MSc Computer Application Technology (Huazhong University), PGD Cs (Mak) and BSc Lib & IS (Mak)	S.Lecturer	10	Permanent
Dr. Raymond Mugwanya	PGD Education (UCT5) Ph.D. Computer Science (UCT), M.Sc. Computing (Liverpool), and B.Sc. Statistics (Mak).	Lecturer	10	Permanent
Dr. Peter Khisa Wakholi	Ph.D. (Bergen), M.Sc. Internet & Database Systems (LSBU),& B.Sc. CS (Mak).	Lecturer	10	Permanent

APPENDIX B: ANNUAL PROGRAMME BUDGET

A	**SUMMARY**		
i	**Tuition Fees**		
	EXPECTED INCOME	Number	**Total(UGX)**
	Total Admission Number	**200**	
	Number of Semesters per year	**2**	
	Tuition Payable per student per Semester		1,512,000
	Total Amount per Semester		302,400,000
	Total Amount per year		604,800,000
	Recess term in first year per student		200,000
	Total amount from Recess term	200x200,000	40,000,000
ii	**DISTRIBUTION**		
	College (33.3%)		201,398,400
	Central Activities (66.7%)		403,101,600
	Total		604,800,000
B	**DETAILED COSTING**		
	INCOME		
	College Collection		201,398,400
	EXPENDITURE		
i	**Teaching Expense**		
	Lecture/tutorial/practical hours	50,000x1920CH	96,000,000
	Bachelor projects supervision/panels *2	200 x 200,000	40,000,000
ii	**Administrative Activities**		
	College Activities (Administration/ Cleaning, Furnishing, Wear and Tear)	-	50,398,400
iii	**Teaching Materials**		

	Item	Quantity	Amount
	Laboratory Costs & Specialised Software	–	10,000,000
	Stationary & Contingencies	–	5,000,000

Annex C : Contact with the Departmetn of Mathematics.



MAKERERE UNIVERSITY

P.O. Box 7062 Kampala - Uganda
URL: www.cit.mak.ac.ug
URL: http://www.cis.ac.ug

Tel: +256 - 414 - 540 628 / 534 560 / 1-97
Fax: +256 - 414 540 620
twitter: @cocismak
facebook: www.facebook.com/cocismak

WE BUILD FOR THE FUTURE

COLLEGE OF COMPUTING & INFORMATICS TECHNOLOGY
DEPARTMENT OF NETWORKS

DEPARTMENT OF MATHEMATICS
MAKERERE UNIVERSITY
03 FEB 2021
P.O.BOX 7062, KAMPALA

03rd February 2021

The Head of Department,
Mathematics,
CoNAS, Makerere University

Dear Sir/Madam

RE: REQUEST FOR REVISED BACHELORS COURSE CONTENT IN MATHEMATICS TO BE INCLUDED IN COCIS CURRICULUM REVIEW.

I hope that this letter finds you in good health.
I write in request for updated course content for a Mathematic course that has been included in the curriculum in the School of Computing and Informatics Technology of CoCIS. The genesis of this request is the presentation of the curriculum at the quality assurance committee of Makerere University, where we were informed by a committee member, also staff at your school that the content we have included has been revised.

Particular course is **MTH2203 Numerical analysis**, which appears in the first year, second semester of the Bachelor of Science in Software Engineering.

Your response to this request will be appreciated.

Regards,

Kyanda Swaib Kaawaase,
Ag. Head of Networks Department, CoCIS

Content shared by Email.

MAKERERE UNIVERSITY
HEAD. DEPARTMENT OF MATHEMATICS

Annex D : Stakeholder's workshop held on 28th November 2019

MAKERERE UNIVERSITY
College of Computing and Information Sciences, SCHOOL OF COMPUTING AND INFORMATION TECHNOLOGY (SCIT)

Department of Networks

REPORT FOR CURRICULUM REVIEW FOR BACHELOR OF SCIENCE IN SOFTWARE ENGINEERING (REVIEW) & B.SC.DATA COMMUNICATIONS NETWORKS (NEW)

Date: 28th November 2019

Venue: Conference Room, Level 4, Block A, College of Computing and Information Science

Contents

1.0 Prayer:

Opening prayer by Kahuma A. Clare

2.0 Introduction of Participants:

Participants introduced themselves and registered their details as follows:

9.2: List of Participants (Appendix 1)

No.	Name of Participant	Organisation /Position	Signature
1.	Prof. Tonny Oyana	Principal, COCIS	
2.	Assoc. Prof. Gilbert Maiga	Dean, SCIT	
3.	Dr. Swaib Kyanda Kaawaase	Head, NW Department	
4.	Mutegeki Henry	Student leader	
5.	Kahuma A. Clare	Student leader	
1.	Kasumba Robert	Cabral Tech (Alumnus)	
2.	Charles Halonda Funa	PRO, COCIS	
3.	Mabirizi Ahmed	Student (President)	
4.	Dora Bampangana	Grants Officer, COCIS	
5.	Dr. Julianne Sansa-Otim	Senior Lecturer, COCIS	
6.	Tadeo Taremwa	Graduate Student (Alumnus)	
7.	Dr. John Ngubiri	Senior Lecturer, Computer Science Department/Director RISE	
8.	Alex Mwotil	Assistant Lecturer, NW Department	
9.	David Batanda	MTN Uganda	
10.	Kasumba Henry	Head of Mathematics	

		Department, CONAS	
11.	Dr. Joab Ezra Agaba	Lecturer, NW Department	
12.	Kahuuta Godwin	Principal Training Officer, Ministry of ICT & Innovations	
13.	Dr. Drake Mirembe	Senior Lecturer, NW Department	
14.	Ronald Azairwe	Managing Director, PEGASUS	
15.	Dativa Byabagabi Ainembabazi	Pegasus	
16.	Byron Maan	CEO, Barracuda Uganda	
17.	Dr. Agnes Nakakawa	Lecturer, Information Systems Department	
18.	Pearl Arinda	Administrative Assistant, SCIT	
19.	Tumuhairwe Bruno	Student Leader	
20.	Asiimwe Paddy	Assistant Lecturer, NW Department	
21.	Mary Nsabagwa	Assistant Lecturer, NW Department	
22.	Dr. Joyce Nakatumba - Nabende	Lecturer, Computer Science Department	
23.	Olowo Norman	Student	

3.0 Remarks (introduction to the Workshop) by Assoc. Prof. Gilbert Maiga, Dean CIT

(i) Welcomed members to the workshop

(i) Informed members that the aim of the workshop was to provide feedback to the Bachelor of Science in Software Engineering curriculum that is currently being reviewed and to critique the new curriculum (B.SC. Data Communication Networks) being developed by the Department of Networks

(ii) That the proposed/new curriculum is meant to contribute to the graduate training in the Department in terms of teaching and learning. Also feeding into the graduate programme that is already running

(iii) The curricular should prepare students for research

(iv) Thanked the Networks Department staff and Head for the job well done so far

4.0 Welcome Remarks from the College Principal, Prof. Tonny Oyana

1. Welcomed participant to the workshop
2. Recognized the presence of the Dean, SCIT, Assoc. Professor Gilbert Maiga.
3. Thanked the Head of Department, Dr. Swaib Kyanda and his team for the work done
4. Thanked the stakeholders available and asked them to critique the material that was to be discussed and what they would like to see
5. Very encouraged with the forward looking vision and mission of the NW Department. As a leader Principal is supportive and making sure the staff and students succeed. He is committed to solving problems and helping the customers who are mainly the students
6. Congratulations to the Head for the hard work. Reviewing BSSE and developing new DCN was not an easy task
7. 5 G network is going to be a big deal in the near future. COCIS is helping organisations like Telecommunication companies to achieve their goals
8. Networks technology is the way to go. In homes, transport, security etc. sensor technology is key in this era. One of the big things going on is climate change. Network technologies need to step in and see how climate change can be tackled
9. Encouraged staff to learn more and innovate more and Principal is ready to support
10. Hopes that the programs will begin to run next year so that we have some new ideas and initiative on board.
11. Encouraged staff of Networks Department to recreate the programmes in order to create excitement in the department
12. We need research agenda on board. There is need to compete for research resources. The Department of Computer Science and Department of Networks have what it takes to bring in money to Makerere University.

13. Some challenges are staff going out to look for opportunities but they should be encouraged to look within. There are facilities like WIMEA-ICT lab which can support innovations. People should encourage one another. WIMEA-ICT project is doing well and should even apply for funding on NORHED 2.
14. Identify researchers in embedded systems. We need research scientists on board, not just research assistants.
15. Forensic and digital experts are needed on board
16. Participants should strive to solve national challenges in the areas of our expertise
17. project managers are critical – from concept to implementation. Students should take note equip themselves with knowledge of successful implementation of projects. They should be all round, adaptable and flexible.
18. We need network engineers and Dr. Julianne Sansa-Otim, former Head of Networks Department should take lead
19. Where there is demand there should be supply. As this workshop goes on, be mindful of the demand. Some people are smart but have bad attitude.
20. People cultivate dynamism for their work, exploring different angles every time.
21. Encourage students to work hard, do not give them predetermined positions. Improve communication
22. Job market is very competitive. We should have positive attitude. Creative thinking should be driven by market demand.
23. We should recreate ourselves to keep marketable and relevant

4.0 Presentation on the industry perspectives/ needs for B.Sc.Data Communications Networks graduates: Mr. David Batanda – MTN Uganda (Appendix 1)

4.1 Highlights from Presentation

The Discussion was mainly on MPBN; Containers and People)

(i) Each day we generate 2.5 quintillion bytes of data

(ii) 2 years of twitter tweets produce more words than are contained in all the books ever printed

(iii) MPBN:

- Transit layer (fiber of the core)
- IP carrier grade network – designed for the future (5G)

- Segmented based on traffic types – segment routing 6
- Designed for speed (5G)
- Hence compression and not encryption
- Automated service (pay for what you want – QoS)
- High latency
- Dynamic bandwidth management
- AI and machine learning models (programmable network – python)
- Slicing – data categorization
- IP addressing (IoT devices)
- Saas, Paas, Iaas ….
- Everything is now moving to cloud based – everything is now hosted on the cloud
- MTN is looking to build its own cloud service

(iv) Containers Vs virtualization

(v) Data Framework: overall big data and analytics approach for MTN- open source architecture

(vi) Security – zero tolerance to security in MTN. Have tools to detect and stop.

(vii) CIA is about confidentiality, access to the right information

(viii) Phone inscription is key

(ix) Security on mobile money is critical

(x) Analytics to be accurate and trustworthy

(xi) Inscription concepts of data in transit and at rest

(xii) Availability and access – batch management and business continuity frameworks is also key to MTN

(xiii) MTN is always testing its security systems

(xiv) Future of the workplace: Skills for the workplace; Problem solving is very key; communication and listening; customer service and empathy; team work and collaboration. Iteration is key in MTN projects. SCRAM frameworks)

(xv) Open API – startup /entrepreneurship

(xvi) ‘tomorrow belongs to companies and individuals who are approaching education in creative manner

4.2 Comments:

(i) Great presentation, very educative

(ii) We need more collaboration with MTN during course delivery. Through visiting lecturers, graduate trainees and mentorship, research collaborations

(iii) MTN has a framework for collaborations

(iv) There is a gap in the market that the DCN curriculum can address. People skills are critical. Execution oriented, agile graduates are more employable.

5.0 Presentation of the industry perspectives/ needs for Bachelor of Science in Software Engineering graduates: Director - Pegasus, Mr. Ronald Azairwe (Appendix 2)

5.1 Presentation Highlights

(i) Evolution on financial services is key in this era and should be tapped into by graduates as way to provide solution and also to create jobs

(ii) There is a movement where institutions like MTN and individuals can pay taxes to the government and this area needs IT expertise to provide solution

(iii) Good education provides solution to challenges in the environment

(iv) Real education is in how one identifies problems and solves them

(v) Opportunities are every where, the graduates need to be trained to look for them

(vi) What does the market need? Good attitude; dependability; discipline; integrity; innovations; system design; ability to learn and re-learn.

(vii) Get small problems, design them well and create solutions

5.2 Comments:

(i) Exciting presentation

(ii) We should give students a chance to think and produce what they have not been taught.

(iii) The presenters have given us a different perceptive and it is our duty as academicians to improve the curricular in that line

(iv) Students should be taught to think creatively

(v) Methods of delivery of the content should be emphasized

(vi) Point of sale terminals should be considered

(vii) Professors in academia should try and work in the industry for a certain percentage of the time through consultancy etc and this time should be accounted for

(viii) We need to change/recreate how we teach

(ix) How have you overcome the barriers of solving challenges in the country. Get someone at the top so you are not bogged down by the lower cadres that may be benefiting from the challenges

6.0 Presentation of draft curriculum for Bachelor of Science in Software Engineering by Dr. Joab Ezra Agaba – Lecturer, Department of Networks (Appendix 3)

Feedback:

(i) Communication Skills: course needs to be re-engineered. NW can change the course name and get technical lecturers to deliver it. We are interested in functional communication. The new course outline should define what is required. Change the content and material. It should not be generalized for the college.

(ii) Some courses are cross cutting. Can NW department ask at the point of delivery for the customization of that particular course unit. Can we may be change course code so that it is no longer a university wide course and department can own it. The Department to sit again and consider these various submissions

(iii) Object Oriented Programming 1& 2: over splitting courses brings about lazy students. This can be combined with structured programming and taught at once.

(iv) Numeric analysis may not be necessary for Software Engineers. Mathematics for SE 1 & 2 should be merged with Numeric analysis

(v) Program objectives and outcomes need to be re-aligned. An objective should have an outcome.

(vi) Preliquisites: they are not clear. Some are available but not all.

(vii) Page 19: course has 09 objectives and that's abit too much. It is a borrowed course from another department and NW department has no control over it.

(viii) Formal methods should be further reviewed

(ix) Design patterns has been incorporated in software architecture and patterns taking care of reliability and testing of software

(x) Delivery system should not encourage laziness but should also help to nurture 'weak' students into becoming better software engineers

7.0 Presentation of draft curriculum for B.Sc.Data Communications Networks by Dr. Swaib Kaawaase Kyanda, Chair, Head, Department of Networks (Appendix 4)

7.1 Comments:

1. Why was it necessary to introduce the course in Makerere University. This should be the rationale.
2. The document should show the need for graduates in the market

3. References: should be in APA style consistently
4. Page 16: course description has some words that are not necessary.
5. Data analysis and visualization: course content and modules …. Why not call them modules.
6. Fees billing structure is not clear. Should be made more clear.
7. Block chain: Mak if we are to be at the top, there is emerging trends which takes of emerging trends in technology. We may not need a specific course in block chain. But what is taught in emerging trends is to the discretion of the lecturer and the Head of Department. Block chain may not necessarily be a NW course. The core fundamentals of NW will be considered in the emerging trends course unit.
8. Recess term: can CCNA be incorporated. Content is taken care of in the programme but not vender specific such as CNNA
9. Entry requirements: could drop physics as essential. Mathematics cannot be essential and also relevant, should remain essential
10. Mathematics 1: students should come up with applications at the end of the course unit
11. Research methodology course comments and some additional items in the reference list have been sent by email to the HOD by Dr. Nakakawa.

8.0 Wrap Up & the Way forward by Dr. Drake Mirembe, Senior Lecturer, Department of Networks

24. The ecosystem with the industry has been learnt
25. Delivery methods should keep evolving
26. Department of Networks should ensure every 2.5 years we have interaction with stakeholders
27. We should consult via tracer studies to know what is really required, prepare and consult
28. Going forward: NW Department to reconvene and discuss the matters raised working towards having the programs running in September 2020.

9.0: Workshop Programme

TIME	ACTIVITY	Responsible
08:30 – 9:00	Registration of Members	Secretariat
9:00 – 9:10	Opening Remarks (introduction to the Workshop) & invite Principal	Assoc. Prof. Gilbert Maiga, DEAN SCIT
9.10-9.20	Welcome Remarks from the College Principal	Prof. Tonny Oyana, Principal, COCIS
9:20 – 9:50	Presentation of the industry perspectives/ needs for Bachelor of Science in Software Engineering graduates	Director - Pegasus, Mr. Ronald Azairwe
9:50-10:20	Presentation on the industry perspectives/ needs for B.Sc.Data Communications Networks graduates	Mr. David Batanda, MTN-Uganda
10:20-10:40	Tea Break	Secretariat
10:40 – 11:10	Presentation of draft curriculum for Bachelor of Science in Software Engineering	Dr. Joab Agaba, Lecturer, Department of Networks
11:10-11:40	Presentation of draft curriculum for B.Sc.Data Communications Networks	Dr. Swaib Kaawaase Kyanda, Chair, Head, Department of Networks
11:40 – 12:30	Cluster Discussions (did not take place)	
12:30-13:45	Lunch Break	Secretariat
13:45-14-45	Receiving input and feedback from cluster discussions (did not take place, as participants were giving feedback directly after presentations)	Dr. John Ngubiri, Senior Lecturer & Director, RISE
14:45-15:15	Wrap Up & the Way forward	Dr. Drake P. Mirembe, Lecturer, Networks Department
15:30-	Break Tea and end	Secretariat

.

Annex E : Stakeholder's workshop held on 13th January 2020 with CEDAT staff

MAKERERE UNIVERSITY , COLLEGE OF COMPUTING AND INFORMATION SCIENCES, DEPARTMENT OF NETWORKS

MEETING BETWEEN NETWORKS DEPARTMENT AND COLLEGE OF ENGINEERING, DESIGN ART AND TECHNOLOGY (CEDAT) ON CURRICULA DEVELOPMENT / REVIEW FOR B.SC. DATA COMMUNICATION AND NETWORKS & BACHELOR OF SCIENCE N SOFTWARE ENGINEERING HELD OF 13TH JANUARY 2020, 3.00PM IN COCIS SEMINAR ROOM

Agenda

1. Recording of Members / Self Introductions
2. Prayer
3. Receive and Discuss comments from the stakeholders' workshop
4. Receive from CEDAT, the similarities and Differences in the DCN
5. 5.0 Discussions on the Stakeholders Report on BSSE:
6. Timelines
7. AOB

1.0 Recording of Members

1.	Dr. Swaib Kyanda Kaawaase	Chair, NW Department	……………………..
2.	Andrew Katumba	CEDAT	……... ……………..
3.	Tonny Ssetumba	CEDAT	……………………..
4.	Dr. Jonathan Sserugunda	CEDAT	……………………..
5.	Dr. Agaba Joab Ezra	Lecturer, NW Department	……………………..
6.	Mr. Alex Mwotil	Asst. Lecturer NW Dept	…………………….
7.	Ms. Mary Nsabagwa	NW Department	…………………….

Secretariat

1. Dora Bampangana ……………………………………….

2.0 Prayer: by Johnathan Sserugunda

3.0 **Receive and Discuss comments from the stakeholders' workshop**

(i) This is a follow up of the Stakeholders workshop NW had on 28th November 2019

(ii) HOW can we improve DCN programme?

4.0 Receive from CEDAT, the similarities and Differences in the DCN

Year 1 Semester 1

(i) THE STRUCTURE looks okay and similar to Telecommunications engineering apart from some names changing e.g computer literacy is Communication Skills just like the entire university

(ii) Why introduce fiber optic networks this early in Year 1 sem 1. It could come later on in the course probably

(iii) Issues of

(iv) Senate wants fewer undergraduate courses. Undergraduate should be as broad as possible to leave room for specialization at graduate level.

(v) There to programmes (BSSE and DCN) should be very distinct.

(vi) Johnathan to share the Telecommunications Engineering course for further reference

(vii) Semester 1, Year 1: Physical electronics and Introduction to Electrical Engineering are quite represented in the Telecommunications Engineering of CEDAT

(viii) Suggestion: Computer Literacy could get a low number of credits other than making it optional. This makes it mandatory for all students as some students may decide not to do it. Even some who may have not done computer may choose not to do it. Some new entrants may not have done the done any computing at all in secondary school

(ix) Introduction to Fibre Optic Networks: it is introduced too early in the programme and has no follow up course unit.

Year 1 Semester 2:

(i) Computer Programming Fundamentals: intended to be an introductory course. This introduces students to programming discipline.

(ii) In the recess, we hope to introduce more network control systems

(iii) Embedded Systems, students use embedded C

Year 2

(i) Digital Multi-Media: name seems incomplete. Johnathan to share the new curricular for CEDAT for further reference

(ii) Introduction to IoT and also Introduction to Web Development … courses will be queried at Senate, they do not like introducing course units in 2nd or 3rd years

Year 3:

(i) Innovation ad Entrepreneurship: To help students be innovative and create their own jobs

(ii) Emerging trends in data communication networks: for any related current trends

(iii) 10 CUs is always on the higher side, a

(iv) Machine learning things are to do with AI issues

(v) Creating data networks needs to put into perceptive the social views, thus a sociology related course may be required.

(vi) Network Ethics Policies & Standards - makes it look like you are teaching only ethics, title should be modified

Agreed Position:

-How much percentage is this course related to CEDAT courses? HOD to work out a percentage using the courses that Johnathan will share

5.0 Discussions on the Stakeholders Report on BSSE:

1.Pre-requisites: should be available for all course units

2. 9 objectives are abit too much, IST 1101: Foundations of IS and Technology. They should be reduced

3. reliability and testing, testing patterns should be reviewed

4. page 10-11 of the workshop report to be looked into by Joab

5. Pages 11-12, HOD to look into them

6.0 Timelines:

1. Submit the documents to the School by Friday 17th Jan 2020
2. School board to be called ASAP as these programmes have to run next academic year

7.0 A.O.B

There being no other business, the meeting ended at 5.16 PM.