

National Council for Higher Education

Minimum standards for the
courses of Study in

Bachelor of Science in Software
Engineering

2011

1. INTRODUCTION	4
2. ANALYSIS OF THE COURSES THAT ARE TAUGHT AT MAKERERE UNIVERSITY IN UGANDA IN COMPARISON WITH THE SWEBOK STANDARD THEMES	6
3.DETAILED MINIMUM STANDARDS	10
2. INTRODUCTION TO PROGRAMMING	14
3. OPERATING SYSTEMS	15
4. INFORMATION SYSTEMS	16
5. COMMUNICATION SKILLS	17
6. SYSTEMS ANALYSIS AND DESIGN	18
7. OBJECT-ORIENTED ANALYSIS AND DESIGN	19
8. COMPUTER ARCHITECTURE AND ORGANISATION	21
9.PRINCIPLES OF SOFTWARE DEVELOPMENT	22
10. DATABASE SYSTEMS	23
11. DATA STRUCTURES AND ALGORITHMS	24
14 . HUMAN COMPUTER INTERACTION	28
16. OBJECT-ORIENTED PROGRAMMING	30
17. REQUIREMENTS ENGINEERING	32
18.NETWORK APPLICATION DEVELOPMENT	33

19. SOFTWARE ARCHITECTURE	34
20. SOFTWARE METRICS	35
21. SYSTEMS PROGRAMMING	37
22. SOFTWARE RELIABILITY & TESTING	38
23. INFORMATION TECHNOLOGY PROJECT MANAGEMENT	39
25. RESEARCH SKILLS	40
25. MOBILE COMPUTING AND APPLICATIONS	43
26. SOFTWARE ENGINEERING MINI PROJECT	44
27. INDUSTRIAL TRAINING	45
28. BUSINESS LAW	48
29. PROFESSIONAL ISSUES IN COMPUTING	49
30. SOFTWARE DESIGN PATTERNS	50
31. INTERNET PROGRAMMING	52
32. COMPUTER GRAPHICS	53
33. SOFTWARE EVOLUTION	54
34. ENTREPRENEURSHIP AND BUSINESS	56
35. INFORMATION SECURITY	58
36. FINAL YEAR PROJECT	59

1. Introduction

According to the computing curricula guidelines developed by the ACM and IEEE-CS [1] (professional association of computing specialists), the Software Engineering discipline is the systematic development and application of techniques which lead to the creation of correct and reliable computer software. By graduation the students of Software Engineering must therefore:

1. Be well-educated in the fundamental concepts of software engineering so that they able to continue their professional development throughout their careers.
 - Have an appreciation of software engineering theory with consideration of its application in the software engineering practice.
 - Possess an understanding of fundamental mathematics and engineering science consistent with the problem solving abilities of a degreed professional in the software engineering field
 - Be proficient in developing software in a variety of languages, platforms and applications using a methodical approach
 - Be able to participate in the multi-disciplinary process of design and qualification of a prototype
 - Have sufficient knowledge of the Software engineering profession that supports informed and timely career decisions
 - Be innovative and are capable of creating jobs

Secondly Software Engineers must exhibit strong ethical principles and have good interpersonal communication and team skills. Students should therefore be able to:

1. Critically evaluate and possibly act on current ethical issues in the IS field
 - Understand contemporary global and societal issues and their relationship to professional ethics and engineering solutions
 - Appreciate an early knowledge of key computer and study skills that are supportive of lifelong learning
 - Collaborate with other professionals as well as perform successfully at the individual level
 - Communicate effectively with excellent oral, written, and listening skills.
 - Demonstrate persistence, flexibility, curiosity, creativity, risk taking, and a tolerance of these abilities in others.

Currently in Uganda Software Engineering (SE) as a bachelors' programme is offered by only Makerere University. This report analyses the SE themes proposed in the Software Engineering Body of Knowledge (SWEBOK) [2] in comparison with the Makerere University SE curriculum. The result of this analysis is a hybrid document that reflects the minimum standard expectations from a SE graduate from a university in Uganda. The skills and competencies of a software engineering graduate can be outlined in nine core knowledge areas. These are:

- Software Requirements and Design
- Software Construction
- Software Quality Assurance and Testing
- Software Engineering process, tools and methods
- Software Engineering Maintenance and Management
- Mathematics and Systems Foundations of Software Engineering

- Computing Foundations of Software Engineering
- Professional, Business and Social Skills
- Practical Considerations

2. Analysis of the Courses that are taught at Makerere University in Uganda in Comparison With the Swebok Standard Themes

This section provides an analysis of the Makerere University SE curriculum in comparison with the SWEBOK theme areas. Each course unit is grouped in one of the nine knowledge areas. Table 1 shows the number of courses that are taught in each of the knowledge areas. The names of the courses can be found in Appendix 1

Knowledge Area	SWEBOK Standard	Makerere
Software Requirements & Design	6	6
Software Construction	3	8
Software Quality Assurance and Testing	3	2
Software Engineering Process, Tool & Methods	3	5
Software Engineering Maintenance and Management	3	2
Mathematical and Systems Foundations of Software Engineering	3	14
Computing Foundations of Software Engineering		
Professional, Business and Social Skills	5	8
Practical Considerations	1	4

Table 1: Number of courses taught in each knowledge area

From Table 1, it should be noted that in some knowledge areas (e.g. Software Construction, Foundations of Software Engineering and Practical Considerations), a single SWEBOK theme can be too wide to be covered as a single semester course unit. In such cases, more than one course unit is mapped onto a single SWEBOK theme. The total number of course units in the Makerere SE curriculum is 49 which is representative of a four year university engineering programme. In addition each knowledge area is covered with a good emphasis considering the number of courses per knowledge area. From a critical look at the course names (shown as Appendix 1), the following can be seen:

2. A course unit such as Computer literacy is not a relevant foundational course any more considering that its material is covered in the computer studies subject at secondary school and

also the proliferation of computers in everyday living means that by university the students already have the skills that computer literacy gives.

3. Under the professional, business and social skills there is a single industrial training course unit. Considering that SE is a 4 year programme and the collaborative nature of the SE profession, a second industrial training course unit would be beneficial e.g. one in the second year recess term coupled with the mini software projects and the second one in the third year recess term.

• The Proposed Minimum Standard

The data gathered from Makerere University as well as ACM and the SWEBOK has been used to develop the minimum standards that a Software Engineer in Uganda should possess. Table 2 shows the proposed minimum number of courses that should be taught in each of the knowledge areas that were identified. It also proposes the core courses that should be made compulsory. Please note that due to the different nomenclature that is used, the course names that are proposed here should not be taken to be final. The universities are free to choose the actual course names of their choice.

Knowledge Area	Proposed No. of courses	Justification	Proposed courses
Software Requirements & Design	6	Understanding the Software Requirements Process, Elicitation, Analysis & Specification is a critic starting point to successful software projects. Secondly followed with the proper software design is important	<ol style="list-style-type: none"> 1. Systems Analysis and Design 2. Requirements Engineering 3. Software Architecture 4. Human Computer Interaction 5. Object oriented Analysis and Design 6. Software Design Patterns
Software Construction	8	A thorough knowledge of the software construction Tools and Techniques as well as construction management is necessary to translate designs into actual software solutions	<ul style="list-style-type: none"> • Structured Programming • Object-Oriented Programming • Network Application Development • Embedded Systems Software • Real-Time and Embedded Applications • Systems Programming • Mobile Computing and Applications • Internet Programming
Software Quality Assurance and Testing	2	An appreciation of the Software Quality Assurance Process as well as Software Test metrics & Processes plays a big role in ensuring software quality	<ol style="list-style-type: none"> 1. Software Metrics 2. Software Reliability and Testing
Software	2	Its critical to have an idea of	<ol style="list-style-type: none"> 1. Principles of Software

Engineering Process, Tools and Methods		the SE Process, Tools & Methods that are available in building SE solutions	Development 2. Computer Graphics
Software Engineering Maintenance & Management	3	SE Project Planning should be understood for purposes of managing new software project, while the SE maintenance process should be appreciated in order to keep already existing software functioning	1. Software Evolution 2. IT Project Management and Development 3. Information Security
Mathematical and Systems Foundations of Software Engineering	3	Provide students with Mathematical and Systems Engineering Foundations	1. Discrete Mathematics 2. Formal Methods 3. Software Systems Engineering
Computing Foundations of Software Engineering	6	Provide students with Computing Foundations	i) Information Systems ii) Algorithms & Data Structures iii) Database Systems iv) Computer Architecture and organisation v) Operating Systems vi) Computer Networks and Data Communication
Professional, Business & Social Skills	6	Students must get skills and knowledge in the professional, business and social domains in order to ably present their solutions. Although each of these fields in broad, nine course units are sufficient as core.	1. Communication Skills & Report writing 2. Business Law 3. Research Skills 4. Entrepreneurship and business 5. Internship 6. Social and professional issues in Computing
Practical Considerations	2	To provide the students to apply the knowledge obtained in all the other knowledge areas	1. Professional Software Engineering Mini Practical Project 2. Final Year Software Engineering project

Table 2: The Proposed core courses

3.Detailed Minimum Standards

Thirty Eight (38) courses have been identified as core areas that should be included within the minimum standards for the Software Engineering graduate. Each University is free to include more courses, over and above these ones. They can even give them unique names (not necessarily the proposed names) but the content covered in all these courses must be taught to the students. The details of the courses that form the minimum standards in the nine knowledge areas are found below organised sequentially in the order in which they should be covered during the four year programme.

Year I Semester I							
	Name	LH	PH	TH	CH	CU	Type
	Discrete Mathematics	30	-	30	45	3	Core
	Introduction to Programming	40	40	-	60	4	Core
	Operating Systems	45	-	-	45	3	Core
	Information Systems	45	-	-	45	3	Core
	Communication Skills	30	-	30	45	3	Must know
Total							

Year I Semester II							
	Name	LH	PH	TH	CH	CU	Type
	Systems Analysis and design	30	-	30	45	3	Core
	Object oriented Analysis and Design	25	-	40	45	3	Core
	Computer Architecture and Organization	30	-	30	45	3	Core
	Principles of Software development	45	-	-	45	3	Core
	Database Systems	45	30	-	60	4	Core
Total							

Year II Semester I							
	Name	LH	PH	TH	CH	CU	Type
	Data Structures and Algorithms						
	Computer Networks and Data Communications	40	40	-	60	4	Core

	Formal methods	35	-	20	45	3	Core
	Human Computer Interaction	30	-	30	45	3	Core
	Software Systems Engineering	30	-	30	45	3	Core
Total							

Year II Semester II							
	Name	LH	PH	TH	CH	CU	Type
	Object oriented Programming	28	44	-	60	4	Core
	Requirements engineering	45	-	-	45	3	Core
	Network Application Development	30	30	-	45	3	Core
	Software Architecture	45	-	-	45	3	Core
	Software Metrics	30	-	30	45	3	Core
Total							

Year III Semester I							
	Name	LH	PH	TH	CH	CU	Type
	Systems Programming	45	30	-	60	4	Core
	Software reliability and testing	42	36	-	60	4	Core
	IT project management and Development	40	10	-	45	3	Core
	Research Skills	30	-	30	45	3	Core
Total							

Year III Semester II							
	Name	LH	PH	TH	CH	CU	Type
	Embedded systems software	45	30	-	60	4	Core
	Mobile computing and Applications	45	30	-	60	4	Core
	Professional SE Min Project	-	120	-	60	4	Core
	Industrial Training	-	120	-	60	4	Core
Total							

Year IV Semester I							
	Name	LH	PH	TH	CH	CU	Type
	Real time and embedded	45	30	-	60	4	Core

	applications						
	Business Law	40	-	40	60	4	Core
	Professional Issues in Computing	35	-	20	45	3	Core
	Software Design Patterns	30	30	-	45	3	Must Know
	Internet Programming	25	40	-	45	3	Must Know
Total							

Year IV Semester II							
	Name	LH	PH	TH	CH	CU	Type
	Computer Graphics	25	40	-	45	3	Core
	Software evolution	39	-	12	45	3	Core
	Entrepreneurship and Business	39	-	12	45	3	Core
	Information Security	35	20	-	45	3	Core
	Final Year Project	-	120	-	60	4	Core
Total							

1. Discrete Mathematics

Course Name	DISCRETE MATHEMATICS
Course Level	Level I Semester I
Course Credit	3CU
Contact hours	45 Hours

Brief Course Description

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

Course Objectives

The course aims to provide students with:

- i) Knowledge about operations associated with sets, functions, and relations with examples
- ii) Understanding of the basic counting principles, including uses of diagonalization and the pigeonhole principle
- iii) Understanding of which proof is best for a given problem and the basic structure of each proof technique giving examples
- iv) The ability to compute permutations and combinations of a set, and interpret the meaning in context of particular applications
- v) The knowledge require to Analyze a problem to create relevant recurrence equations or to identify important counting questions
- vi) An appreciation of the effect of AND, OR, NOT and EOR operations on binary data.

Learning Outcomes

Upon successful completion of the course, students shall be able to:

- i) Explain the basic concepts of discrete structures and appreciate their importance as they apply to computing
- ii) Manipulate formulae involving sets, integers, reals and functions of such quantities
- iii) Solve simple problems involving sets, functions, graphs and trees
- iv) Construct sound logical arguments, including use of induction
- v) Appreciate the way that discrete mathematics can assist problem solving

Course Outline

- | | |
|---------------------------------------|----------|
| • Introduction to Logic and Proofs | 3 hours |
| • Fundamental Structures | 4 hours |
| • Boolean Algebra | 6 hours |
| • Basic Logic | 3 hours |
| • Digital Logic | 6 hours |
| • Elementary Number Theory | 4 hours |
| • Basics of Counting | 4 hours |
| • Case study and Tutorials (30 hours) | 15 hours |

Mode of Delivery

Lectures, Tutorials and Group work

Course Assessment

- Tests 20%
- Assignments 20%
- Examination 60%

2. Introduction to Programming

Course Name	INTRODUCTION TO PROGRAMMING
Course Level	Level I Semester I
Course Credit	4CU
Contact hours	60 Hours

Brief Course description

The course provides a strong base in the principles and practice of structured programming. A high level programming language such as C is used to explain the principles of programming and provide students with hands on practical skills. Areas covered include program structure, data structures, syntactical and semantic correctness, planning and segmentation in programming as well as working with data files.

Course Objectives

The course aims to provide students with:

- Knowledge about the various programming languages
- Knowledge in basic programming concepts
- Comprehensive knowledge about structured programming
- Knowledge in planning and organization of programming projects
- Techniques of evaluating syntactic and semantic correctness of a computer program
- Strong practical foundation in programming

Learning Outcomes

Upon Completion of the course, the students shall be able to:

- Explain the key differences between the various programming languages
- Demonstrate understanding about the basic programming concepts
- Build software using a functional programming language such as C
- Plan and organize a programming project
- Evaluate a computer program for syntactic and semantic correctness

Course Outline

This course covers the following topics:

- | | |
|--|---------|
| i) Introduction to programming languages | 4 hours |
| ii) Program structure | 2 hours |
| iii) Variables and Operators | 2 hours |
| iv) Conditional statements | 5 hours |
| v) Looping statements | 5 hours |

vi) Arrays and strings	5 hours
vii) Functions	3 hours
viii) Advanced data types	3 hours
ix) Pointers	3 hours
x) Dynamic memory allocation and dynamic structures	3 hours
xi) Working with files	3 hours
xii) GUI	2 hours
xiii) Practical sessions (40 hours)	20 hours

Mode of Delivery

Lectures and Practicals

Course Assessment

• Tests	20%
• Practicals	20%
• Examination	60%

3. Operating Systems

Course Name	OPERATING SYSTEMS
Course Level	Level I Semester I
Course Credit	4CU
Contact hours	60 Hours

Course Description

This course introduces learners to the services of operating systems. It exposes them to details on how an operating system controls all the processes of a computer system including those of peripheral devices.

Course Objectives

The aims of this course are to:

- Create understanding about the role of operating systems in the management of computer system resources
- Provide in-depth knowledge of operating system processes and principles
- Enable students to understand computer system performance

Learning Outcomes

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

- Demonstrate understanding about the role of an operating system in the management of system resources
- Explain the principles that govern system operations to improve system performance
- Measure the performance of computer systems

Course Outline

- Operating System concepts **3 hours**

- Structure of an operating system **6 hours**
- System processes **9 hours**
- Threads **6 hours**
- Deadlocks **6 hours**
- Memory management **9 hours**
- Device management **6 hours**

Mode of Delivery

Lectures and Tutorials

Course Assessment

- i) Tests 20%
- ii) Assignments 20%
- iii) Final examination 60%

4. Information Systems

Course Name	INFORMATION SYSTEMS
Course Level	Level I Semester I
Course Credit	3CU
Contact hours	45 Hours

Brief Course Description

This course provides students an overview of the value of information systems within an organization. The course also offers students a chance to examine case studies of IS successes and failures and to explore real-world Information management issues.

Course Objectives

The course aims to provide students with:

- i) Understanding of the different types of information systems and the important role they play in a modern organization
- ii) Knowledge on information systems development and Management
- iii) Understanding of the need for information systems security , audit and ethics
- iv) Understanding of how to manage and leverage an information system for an organization

Learning Outcomes

Upon successful completion of the course, a student shall be able to:

- i) Describe the different types of information systems and the important role they play in a modern organization
- ii) Demonstrate knowledge and skills in information systems development and Management
- iii) Explain information systems security , audit and ethics
- iv) Explain how to manage and leverage an information system to meet the organization goals

Course Outline

- Information Systems Concepts 3 hours

- Information systems infrastructure 12 hours
- The different categories of information systems 7 hours
- Information systems development and management 8 hours
- The role of Information Systems in an organization; 3 hours
- Information systems security 6 hours
- Information systems Audit 3 hours
- Information systems ethics and Crime 3 hours

Mode of Delivery

Lectures, Group work

Course Assessment

- Tests 20%
- Assignments 20%
- Examination 60%

5. Communication Skills

Course Name	COMMUNICATION SKILLS
Course Level	Level I Semester I
Course Credit	3CU
Contact hours	45 Hours

Brief Course Description

This course provides students with skills to effectively communicate with their peers, subordinates and superiors. These skills include verbal, written, and non-verbal gestures. It improves their command of the English language in the areas of reasoning, writing and other expressions.

Course Objectives

The course aims to equip students with skills in:

- Reading, writing and disseminating of information
- Collecting and synthesizing information
- Critical thinking and problem solving
- Utilizing the library and other educational resources

Learning Outcomes

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

- i) Effectively read, write and disseminate information
- ii) Collect and synthesize information
- iii) Generate solutions using critical thinking and problem solving skills
- iv) Utilize the library and other educational resources

Course Outline

The topics covered are in the broad areas of:

- i) Interpersonal skills **6 hours**

ii) Writing skills	9 hours
iii) Reading skills	6 hours
iv) Presentation skills	6 hours
v) Examination skills	3 hours
vi) Tutorials and Presentations (30 hours)	15 hours

Mode of Delivery

Lectures, Tutorials, Presentations

Course Assessment

• Test	20%
• Assignment	20%
• Final examination	60%

6. Systems Analysis And Design

Course Name	SYSTEMS ANALYSIS AND DESIGN
Course Level	Level I semester II
Course Credit	3CU
Contact hours	45 Hours

Brief Course Description

The course uses structured systems analysis and design techniques to develop and document effective computer based information systems. It focuses on the tools and techniques used for analysis and design of systems to meet the increasing need for information within organizations.

Course Objectives

The course aims to provide students with understanding and skills on:

- The systems concepts and the system Development Life Cycle
- The role of the System Analyst in the systems development life cycle
- The techniques of requirements elicitation, specification, and analysis of information systems
- Data modeling in information systems development
- Process modeling in information systems development
- Information system development and maintenance

Learning Outcomes

At the end of the course, students shall be able to:

- Explain the systems theoretical concepts and the system Development Life Cycle
- Explain the role of the System Analyst in the systems development life cycle
- Describe and apply the techniques of requirements elicitation, specification, and analysis to information systems development
- Create a data model for an information systems development project
- Create a process model for an information systems development project
- Explain how Information system construction and maintenance is done

Course Outline

The major topics covered include the areas of:

i) Systems theory	2 hours
ii) The systems development life cycle	3 hours
iii) The role of the systems analyst	1 hour
iv) Information Systems Requirements Analysis	3 hours
v) Process modeling with data flow diagrams	6 hours
vi) Logic and process specification	6 hours
vii) Data modeling with entity relationship diagrams	6 hours
viii) System construction, Implementation and maintenance	3 hours
ix) Case study and Tutorials (30 hours)	15 hours

Mode of Delivery

Lectures, Case studies, Tutorials, Seminars, Group work

Course Assessment

• Tests	20%
• Assignments	20%
• Examination	60%

7. Object-Oriented Analysis And Design

Course Name	OBJECT-ORIENTED ANALYSIS AND DESIGN
Course Level	Level I Semester II
Course Credit	3CU
Contact hours	45 Hours

Brief Course Description

The course covers the important themes of object orientation and UML modeling, critical for the design and development of systems using the object oriented programming languages. Particular emphasis is laid on structure and behavioral modeling as well as object interactions.

Course Objectives

The course aims to provide skills on:

- The concepts and advantages of object oriented analysis, design and its application to systems development.
- Basic object-oriented concepts of; abstraction, encapsulation, inheritance, hierarchy, modularity, and polymorphism
- Creating models used in object oriented programming languages such as C++ and Java
- Iterative, use case-driven process to the development of a robust design model
- Use of UML as a tool or language to represent the design model

Learning Outcomes

Upon Completion of the course, students shall be able to:

- i) Explain the concepts and advantages of object oriented analysis and design for application systems development.

- ii) Apply Object Oriented concepts such as abstraction, encapsulation, inheritance, hierarchy, modularity, and polymorphism to the development of robust systems models
- iii) Represent design models using the Unified Modeling Language

Course Outline

The course covers the following main themes and associated topics:

- i) Objects and classes 2 hours
- ii) Principles of the Object orientated paradigm (Abstraction, encapsulation, inheritance and hierarchy, modularity and polymorphism) 6 hours
- iii) UML modeling and design
 - a. Use case diagrams 6 hours
 - b. Structural diagrams. 6 hours
 - c. Interactions diagrams 6 hours
 - d. Deployment diagrams 3 hours
 - e. Component Diagrams 3 hours
- ii) UML relations: associations, aggregation, composition 3 hours
- iii) Tutorial and Case Study 20 hours

Mode of Delivery

Lectures, Case studies, Tutorials and Group work

Course Assessment

- Tests 20%
- Assignment 20%
- Examination 60%

8. Computer Architecture And Organisation

Course Name	COMPUTER ARCHITECTURE AND ORGANISATION
Course Level	Level I Semester II
Course Credit	3CU
Contact hours	45 Hours

Brief Course Description

This course introduces students to the organization and architecture of a computer. It explores the composition of a computer including its design, controlling functionality, processing and memory management.

Course Objectives

The course aims to:

- Introduce the concepts in computer organization
- Explain the architectural features of a modern computer
- Provide understanding of the principles governing the interaction between hardware and software components
- Describe the relationship between the different components and their roles in the operations of the computer

Learning Outcomes

On completion of this course, the student shall be able to explain the:

- i) Relevant concepts in computer organization
- ii) Architectural features of a modern computer system including its design and development
- iii) Principles governing the interaction between hardware and software components
- iv) Relationship between the different components and their roles in the operations of the computer

Course Outline

The major topics covered include:

- | | |
|--|-----------------|
| i) Digital logic | 6 hours |
| ii) Memory management | 6 hours |
| iii) Computer processor architectures | 6 hours |
| iv) Machine Language and CPU Instructions sets | 6 hours |
| v) Instruction processing | 3 hours |
| vi) I/O Programming | 3 hours |
| vii) Tutorials (30 hours) | 15 hours |

Mode of Delivery

Lectures and Tutorials

Course Assessment

- Tests 20%

- Assignment 20%
- Final examination 60%

9.Principles Of Software Development

Course Name	PRINCIPLES OF SOFTWARE DEVELOPMENT
Course Level	Level I Semester II
Course Credit	3CU
Contact hours	45 Hours

Brief Course description

The course introduces students to principles in software development, from conception of the desired software through to the final manifestation of the software. Principles of software development are basic truths, rules or assumptions about software engineering that hold regardless of the technique, tool or language selected.

Course Objectives

The course aims to provide students with understanding and skills in:

- The software development process and its core activities
- Requirements analysis of a simple software system
- Managing a software development process
- Designing, implementing and deploying a software product
- Software documentation
- Validation, verification and testing a software Product

Learning outcomes

Upon successful completion, the student shall be able to:

- Demonstrate knowledge of the software development process
- Perform requirement analysis of a simple software systems
- Manage a software development process
- Design, implement and deploy a software product
- Demonstrate mastery of at least one tool of software documentation
- Work in a team to design, develop, test and validate a software product

Course Content

Topics covered include:

- | | |
|---|---------|
| i) Introduction to software engineering | 3 hours |
| ii) Phases in the development of software | 5 hours |
| iii) Software project management | 5 hours |
| iv) Requirements analysis and specification | 5 hours |
| v) Software design | 5 hours |
| vi) Software Documentation | 3 hours |
| vii) User Interface design and adaptation | 3 hours |
| viii) Software modularization | 5 hours |
| ix) Software quality assurance | 8 hours |

x) Introduction to software evolution

3 hours

Mode of Delivery

Lectures

Course Assessment

- Tests 20%
- Assignment 20%
- Examination 60%

10. Database Systems

Course Name	DATABASE SYSTEMS
Course Level	Level I Semester II
Course Credit	4CU
Contact hours	60 Hours

Brief Course Description

This course introduces the student to the basic theoretical and practical concepts of a database, its setup, implementation, use and maintenance in a typical business organization.

Course Objectives

The course aims to provide students with:

- The theory of database systems and data modeling
- Practical skills in developing secure relational databases
- Knowledge to formulate queries in databases
- Knowledge of at least one DBMS
- Skills to import and export data to and from different applications

Learning Outcomes

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

- i) Demonstrate understanding of the basic concepts of database systems and data modeling
- ii) Develop secure relational databases
- iii) Formulate queries in databases and to use at least one DBMS
- iv) Import and export data to and from different applications

Course Outline

The topics to be covered include:

- | | |
|--------------------------------------|----------------|
| i) Introduction to database systems | 3 hours |
| ii) Database construction principles | 6 hours |
| iii) Database architecture | 3 hours |
| iv) Data modeling | 9 hours |
| v) Database querying | 6 hours |
| vi) Database Security | 6 hours |

vii) Database Managements Systems	6 hours
viii) Distributed databases	6 hours
ix) Practical lessons (30 hours)	15 hours

Mode of Delivery

Lectures, case studies and Practicals

Course Assessment

- Test 20%
- Practical 20%
- Final examination 60%

11. Data Structures And Algorithms

Course Name	DATA STRUCTURES AND ALGORITHMS
Course Level	Level II Semester I
Course Credit	4CU
Contact hours	60 Hours

Course Description

This course exposes students to data structures and corresponding algorithms. It also equips them with the skills to measure the performance of an algorithm as well as the techniques on how they can be further improved.

Course Objectives

This course aims at equipping students with knowledge about:

- The different types of data structures and identify their appropriateness
- Algorithms and their different representations in terms of mathematical functions or formulae
- The performance and management of simple algorithms
- Cost-effective algorithms

Learning Outcomes

On completion of course, the student shall be able to:

- Use the different types of data structures
- Perform an algorithm analysis and represent algorithms in terms of mathematical functions
- Evaluate the performance and management of simple algorithms
- Develop cost-effective algorithms

Course Outline

The topics covered in this course include:

i) Elementary data structures	6 hours
ii) Advanced data structures	10 hours
iii) Trees, graphs, networks, functions, signals	8 hours
iv) Sorting and searching	5 hours
v) Analysis of algorithms	5 hours
vi) Recursion	3 hours

- vii) Algorithm performance and optimization
- viii) Practicals (30 hours)

8 hours
15 hours

Mode of Delivery

Lectures, tutorials and practicals

Course Assessment

- i) Test 20%
- II) Practical 20%
- III) Final examination 60%

Computer Networks and Data Communication

Course Name	COMPUTER NETWORKS AND DATA COMMUNICATION
Course Level	Level II Semester I
Course Credit	4CU
Contact hours	60 Hours

Brief Course Description

This course introduces concepts of computer networks within the context of the Internet. It covers fundamental principles, elements and protocols of computer networks and data communication.

Course Objectives

The aims of this course are:

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

Learning Outcomes

By the end of this course, the student shall be able to:

- Explain the underlying principles of data communication and networks
- Explain the operation of a range computer networking applications such as email, web, and peer-to-peer file-sharing
- Relate the architecture of the Internet to the underlying design principles
- Illustrate the operation and evaluate the performance of common routing protocols, queuing mechanisms, and congestion control mechanisms
- Develop elements of a network such as gateways and routers that conform to IETF standards with acceptable levels of simplification

Course Outline

- i) OSI and TCP/IP models 7 hours

ii) Transmission of Digital Data and transmitting media	7 hours
iii) Signals, Modulation & Multiplexing	6 hours
iv) Compression and decompression	3 hours
v) Introduction to client-server computing specifically web applications	7 hours
vi) Network management	4 hours
vii) Wireless networks	3 hours
viii) Network Security	3 hours
ix) Practical Lessons/group work (40 hours)	20 hours

Mode of Delivery

Lectures, Practicals and Group work

Course Assessment

• Tests	20%
• Practicals	20%
• Examination	60%

.Formal Methods

Course Name	FORMAL METHODS
Course Level	Level II Semester I
Course Credit	3CU
Contact hours	45 Hours

Brief Course Description

This course presents some formal notations that are commonly used for the description of computation and computing systems, for the specification of software and for mathematically rigorous arguments about program properties.

Course Objectives

The aims of this course are:

- To introduce students to the mathematical notation and terminology used in formal specification
- To equip students with knowledge of the fundamental principles in formal methods including laws and theorems
- To equip students with an understanding of techniques and procedures required to solve practical problems
- To impart programming skills by writing numerical programs required to solve the practical problems mentioned above

Learning Outcomes

Upon completion of this course students shall be able to:

- Demonstrate knowledge including the mathematical notation and terminology of formal methods

- ii) Describe the fundamental principles including laws and theorems arising from formal methods concepts
- iii) Apply formal techniques and procedures to solve practical problems
- iv) Demonstrate programming skills by writing numerical programs, (e.g. Matlab) to solve practical problems

Course Outline

The topics to be covered in this course include:

- | | |
|--|----------|
| i) Predicate Logic Specification | 8 hours |
| ii) Tools and systems of formal notations | 9 hours |
| iii) Algebraic Specification | 10 hours |
| iv) Other Topics e.g. Statecharts, Automatic program synthesis | 8 hours |
| v) Tutorials (20 hours) | 10 hours |

Mode of Delivery

Lectures and tutorials

Course Assessment

- | | |
|---------------|-----|
| • Tests | 20% |
| • Assignments | 20% |
| • Examination | 60% |

14 . Human Computer Interaction

Course Name	HUMAN COMPUTER INTERACTION
Course Level	Level II Semester I
Course Credit	3CU
Contact hours	45 Hours

Course Description

The course helps students to appreciate the delicate and divergent nature of human and computer interactions. It covers the dynamics of user interface designs including their evaluations as well as usability issues for both software and related devices.

Course Objectives

The aims of this course are equip students with knowledge and skills:

- i) To appreciate established human-computer interaction approaches and techniques.
- ii) To design user interfaces
- iii) To use appropriate software evaluation techniques and heuristics

Learning Outcomes

On completion of this course, the student shall be able to:

- i) Apply established human-computer interaction approaches and techniques
- ii) Produce conceptual and physical designs using prototyping methods
- iii) Evaluate software interfaces using appropriate heuristics

Course Outline

- | | |
|--|-----------------|
| • Human Computer Interaction concepts | 5 hours |
| • User Interfaces | 5 hours |
| • Design principles of HCI | 5 hours |
| • Evaluation paradigms, frameworks, heuristics, usability testing | 5 hours |
| • Software posture, flow, navigation, Interaction, language and attributes | 5 hours |
| • Human and social factors | 5 hours |
| • Practicals/ Tutorials (30 hours) | 15 hours |

Mode of Delivery

Lectures, Practicals/ Tutorials

Course Assessment

- | | |
|---------------------|-----|
| • Test | 20% |
| • Assignment | 20% |
| • Final examination | 60% |

15. Software Systems Engineering

Course Name	SOFTWARE SYSTEMS ENGINEERING
Course Level	Level II Semester I
Course Credit	3CU
Contact hours	45 Hours

Brief Course Description

This course examines software as an engineering discipline. It explains the importance of systems engineering and elaborates the related concepts of systems and modern engineering. The emphasis is on complex software systems, requiring a high degree of rigor and formality.

Course Objectives

The aims of this course are:

20. To introduce students to the systems engineering approach to software systems
21. To equip students with analytical skills so they can evaluate the factors leading up to system complication and complexity, the causes of a systems failure, the importance of stakeholders involvement in a systems engineering project and using input-output diagramming
22. To enable students appreciate the importance of systems engineering important as well as the difference between process science and engineering
23. To impart knowledge of a systems engineering project in terms of the balance of demands, choice and constraints
24. To equip students with an understanding of the lag involved in controlling a system as well as the differences between a generic model of systems engineering and a software specific model

Learning Outcomes

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

1. Classify new software product developments as: fault correction, enhancements, new but similar products, radically different, revolutionary or iconoclastic products
2. Analyse the causes of a systems failure
3. Identify and evaluate the relationships of the factors leading up to system complication and complexity
4. Answer the question 'why is systems engineering important?'
5. Define the difference in process between science and engineering
6. Identify and analyse the importance of stakeholders involved in a systems engineering project
7. Classify a systems engineering project in terms of the balance of demands, choice and constraints;
8. Analyse a system using input-output diagramming
9. Estimate the lag involved in controlling a system
10. Evaluate the differences between a generic model of systems engineering and a software specific model

Course Content

- | | |
|---|---------|
| 1. Characteristics of systems engineering | 3 hours |
| 2. Challenges in a systems engineering project | 3 hours |
| 3. Scope of a systems engineering problem | 3 hours |
| 4. Identify the stakeholders and other factors that shape the system requirements | 4 hours |

5. Emergent system properties	3 hours
6. Identifying the real problem which the system is intended to solve	4 hours
7. Technological, operational and economic considerations in the design process	4 hours
8. Interactions between a system and its environment	3 hours
9. Integrating the system with existing systems	3 hours
10. Tutorials (30 hrs)	15 hours

Mode of Delivery

Lectures, Tutorials, Seminars, Group work

Course Assessment

1. Tests	20%
2. Take-home assessments	20%
3. Examination	60%

16. Object-Oriented Programming

Course Name	OBJECT-ORIENTED PROGRAMMING
Course Level	Level 2
Course Credit	4CU
Contact hours	60 Hours

Course Description

The object oriented programming paradigm has been used in many projects to build complex software systems. This course provides students with the opportunity to acquire skills for building software based on the object oriented programming paradigm. C++ and/or Java are some of the languages used in this course.

Course Objectives

The course aims to enable students to:

- Understand and use the Object Orientation Paradigm for software development
- Develop software using a programming language like C++ and/or Java
- Understand the use of basic Java features in a working program
- Understand the use of the following advanced features of inheritance, encapsulation, overloading, polymorphism, abstract classes and interfaces in a program
- Develop knowledge in the use of GUI-based event-driven programming in a working program assignment utilizing GUI components, event-listeners and event-handlers

Learning Outcomes

Upon successful completion of the course, students shall be able to:

- Demonstrate the understanding and application of Object oriented design techniques for software development using a programming language like C++ and Java
- Demonstrate correct use of the basic features in a working program: objects, classes, methods, IO handling, decisions and iterations

- Demonstrate correct use of the following advanced features in a working program: inheritance, encapsulation, overloading, polymorphism, abstract classes and interfaces
- Demonstrate knowledge of GUI-based event-driven programming in a working program assignment utilizing GUI components, event-listeners and event-handlers

Course Outline

Topics include:

- | | |
|---|--------|
| • Theory of the object oriented paradigm | 4hours |
| • Data types, inheritance, exceptions, templates and patterns | 3hours |
| • Object Oriented programming languages (e.g. C++ and Java) | 8hours |
| • Encapsulation, inheritance, and polymorphism | 4hours |
| • Abstract data types, concrete data types and subtypes | 3hours |
| • Virtual functions and operator overloading and pointers | 4hours |
| • Object Oriented computing in a distributed systems, applets | 3hours |
| • Classes, interfaces and packages | 4hours |
| • Selection and application of current design and development tools | 4hours |
| • Practical Lessons (44 hours) | |
| 22hours | |

Mode of Delivery:

Lectures, Tutorials, Practicals, Group work

Course Assessment:

- | | |
|------------------|-----|
| i) Tests | 20% |
| ii) Practical | 20% |
| iii) Examination | 60% |

17. Requirements Engineering

Course Name	REQUIREMENTS ENGINEERING
Course Level	Level 4
Course Credit	3CU
Contact hours	45 Hours

Course Description

Requirements engineering is a software engineering process which covers all of the activities involved in discovering, documenting and maintaining a set of requirements for a computer-based system. It provides students with principles, tools techniques for requirements elicitation, specification, modeling, validation and management.

Course Objectives

The course aims to provide students with:

- i) Knowledge about principles, tools and techniques for requirements elicitation, specification analysis and modeling
- ii) Understanding of the role of requirements in system development and maintenance
- iii) Skills in specifying requirements for Real Systems

- iv) Knowledge about managing changes to requirements as the system is being developed and put into use.

Learning Outcomes

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

- Apply the principles, tools, and techniques for requirements elicitation, specification, and analysis and modeling to a software engineering problem
- Demonstrate the role of requirements in system development and maintenance
- Explain the difficulties of specifying requirements for real systems, as well as effective methods tools and techniques.
- Manage changes to requirements as the system is being developed and put into use

Course Outline

- Software Requirements Concepts, 3hours
- Requirements Elicitation 6hours
- Software Requirements Analysis, 6hours
- Software Requirements Specifications, 6hours
- Software Requirements Tools, 6hours
- Software Requirements Verification, 6hours
- Software Requirements Engineering Management, 6hours
- Developing a Successful Software Requirement . 6hours

Mode of Delivery

Lectures, Case studies, Group work

Course Assessment

i) Tests	20%
ii) Take-home assessments	10%
iii) Examination	70%

18. Network Application Development

COURSE NAME	NETWORK APPLICATION DEVELOPMENT
COURSE LEVEL	LEVEL 2
COURSE CREDIT	3CU
CONTACT HOURS	45 HOURS

Course description

The course introduces students to the design and implementation of network-based applications, demonstrated using Object-Oriented Programming techniques at both the application layer and the transport layer of the TCP/IP protocol stack.

Course Objectives

This course is designed to enable students to:

- i) Understand technologies and protocols supporting computer communication networks
- ii) Understand the technologies and architecture supporting the internet
- iii) Learn network based programming methodologies, languages, tools and standards.
- iv) Acquire Knowledge and Skills on Advanced Network socket programming concepts

Learning Outcomes

Upon completion of this course the students shall be able to

- i) Explain technologies, protocols and architectures that support computer communication networks
- ii) Explain the internet technologies and its architecture
- iii) Explain network based programming methodologies, languages, tools and standards.
- iv) Set up Network sockets using a Structured or an Object oriented programming language
- v) Program and manage a computer network using modern techniques and tools.

Course Outline:

Topics include:

• Networking Fundamentals,	3hours
• layered protocol architectures and service models	3hours
• Design principles for network-based applications	3hours
• HTTP and HTTPS protocols	3hours
• A history and Structure of the Internet	3hours
• connectionless and connection-oriented services	3hours
• Circuit switching and Packet switching fundamentals	3hours
• Network socket options and socket programming	6hours

- Principles of information security in network-based applications 3hours
- Practicals (30 hours) hours 15

Mode of Delivery:

Lectures, Practicals/ Group work

Course Assessment:

- i) Tests 20%
- ii) Practical 20%
- iii) Examination 60%

19. Software Architecture

COURSE NAME	SOFTWARE ARCHITECTURE
COURSE LEVEL	LEVEL 2
COURSE CREDIT	3CU
CONTACT HOURS	45HOURS

Course description

Software architecture defines the primary system qualities of performance, modifiability, and security, into a unifying architectural vision. It is an artifact for early analysis to ensure that a design approach yields an acceptable system. This course also explains the work assignments that must be carried out by design and implementation teams

Course Objectives:

This course aims to enable students to:

- understand the latest state-of-the-art software architecture
- understand software architecture system design
- Understand how system’s components are meant to interact with each other.

Course Objectives:

At the end of the course, students shall:

- Demonstrate familiarity with the latest state-of-the-art software architecture
- Be able to explain a software system design
- Explain how system’s components interact with each other
- Describe a software system design for a particular program

Course Content:

The topics to be covered in this course unit are:

- Architectural styles and Patterns 3hours
- Components of architectural design 3hours
- Connectors, components and composition 6hours

- Architectural design guidance and Tools for architectural design 3hours
- Achieving quality goals with architectural styles 3hours
- Formal models and specifications 6hours
- Analyzing software architecture with SAAM 6hours
- Architecture description languages (ADLs) 6hours
- Architecture-based development 3hours
- Patterns in software architecture 3hours
- Reusing software architecture 3hours

Mode of Delivery

Lectures, Case studies, Group work

Course Assessment

- | | |
|---------------------------|-----|
| i) Tests | 30% |
| ii) Take home assignments | 10% |
| iii) Examination | 60% |

20. Software Metrics

Course Name	SOFTWARE METRICS
Course Level	Level 4
Course Credit	3CU
Contact hours	45 Hours

Course Description

This course helps students appreciate the role software metrics play in the control and management of the software development process. Software metrics provide a quantitative basis for development and validation of models of the software development process. Effective management of any process requires quantification and measurement.

Course Objectives

The course aims to provide a student with:

- An understanding of the need for software metrics and classification of software metrics
- Knowledge of basic terminology and fundamental facts about software metrics and process models
- Understanding of the importance of software metrics in the control and management of software development process
- Knowledge of the fundamentals of measurement theory
- Understanding of software product metrics , process metrics and measuring management

Learning Outcomes

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

- i) Explain basic terminology and state fundamental facts about software metrics and process models
- ii) Identify essential elements of given metrics describe interrelationships among its various elements and describe circumstances or environment in which its use is appropriate

- iii) Describe the importance of software metrics in the control and management of the software development process.
- iv) Understand the foundations of measurement theory and models of software engineering measurement
- v) Appreciate software products metrics, software process metrics and measuring management.

Course Content:

The course is composed of the following basic modules:

- | | |
|---|---------|
| i) Need and definition of Software metrics, | 3hours |
| ii) Classification and measurement of scales for software metrics | 3hours |
| iii) Measurement theory | 4hours |
| iv) Software product and process measurements | 4hours |
| v) Measuring internal product attributes: size and structure | 4hours |
| vi) Measuring external product attributes: quality, measuring cost and effort, | 4hours |
| vii) Measuring software reliability, software test metrics, object-oriented metrics | 4hours |
| viii) Measurement management | 4hours |
| ix) Tutorials and group work | 15hours |

Mode of Delivery:

Lectures, Tutorials, Group work

Course Assessment:

- | | |
|---------------------------|-----|
| i) Tests | 20% |
| ii) Take-home assessments | 20% |
| iii) Examination | 60% |

21. Systems Programming

Course Name	SYSTEMS PROGRAMMING
Course Level	Level 3
Course Credit	4CU
Contact hours	60 Hours

Course Description

The course trains learners on how to develop codes and algorithms that can evaluate and improve on the performance of an operating system. It requires prerequisite knowledge in programming and operating systems.

Course Objectives

The aims of the course are:

- To acquainted students with at least one operating systems software.
- To provide students with skills in developing codes or algorithms that can manage selected system resources.
- To provide students with scripting knowledge and skills, which can be used to measure the performance of selected components of a computer system.

Learning Outcomes

On completion of this course, the student shall be able to:

- Explain basic concepts in at least one operating systems software.
- Develop codes and algorithms that can manage selected system resources.
- Demonstrate the ability to evaluate the performance of selected components of a computer system.

Course Outline

i) Introduction to the Operating System software	6 hours
ii) Operating system Concepts and parameters	6 hours
iii) File I/O	6 hours
iv) System Processes	6 hours
v) Execution of Programs	3 hours
vi) Threads	6 hours
vii) Pipes and messages	6 hours
viii) Memory Mappings	6 hours
ix) Practicals	15hours

Mode of Delivery

Lectures and Practicals

Course Assessment

- Tests 20%
- Practical 20%
- Final examination 60%

22. Software Reliability & Testing

COURSE NAME	SOFTWARE RELIABILITY & TESTING
COURSE LEVEL	LEVEL 3
COURSE CREDIT	4CU
CONTACT HOURS	60 HOURS

Course description

This course presents an overview of the Software Reliability process, techniques and models. Software Reliability is the probability of failure-free software operation for a specified period of time in a specified environment and is an important factor affecting system reliability. The high complexity of software is the major contributing factor of Software Reliability problems.

Course Objectives

The course aims to provide students with knowledge and skills in:

- The software reliability process
- The various software reliability process growth models
- The various metrics used for software reliability measurement
- The techniques that can be used to improve and predict software reliability
- The various techniques applied to software testing
- The application of a software reliability process model to a given case scenario

Learning Outcomes

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

- Explain the software reliability process
- Describe the various software reliability process growth models
- Describe and explain the various metrics used for software reliability measurement
- Explain the techniques that can be used to improve and predict software reliability
- Describe and Explain the various techniques applied to software testing
- Apply a software reliability process model to a given case scenario

Course content

This course covers the following topics:

- | | |
|--|----------|
| i) The software testing and reliability concepts | 4 hours |
| ii) The software reliability process and its application | 4 hours |
| iii) Factors influencing software reliability | 4 hours |
| iv) Software reliability Engineering techniques | 4 hours |
| v) Software reliability models and Measurement | 6 hours |
| vi) software reliability improvement and prediction Techniques | 6 hours |
| vii) Metrics for software reliability prediction | 6 hours |
| viii) Soft ware System testing | 8 hours |
| ix) Practicals and group work | 18 hours |

Mode of Delivery

Lectures, Practicals and Group work

Course Assessment

- Tests
20%
- Practicals
20%
- Examination
60%

23. Information Technology Project Management

Course Name	INFORMATION TECHNOLOGY PROJECT MANAGEMENT
Course Level	Level 3
Course Credit	3CU
Contact hours	45

Course Description

The module defines the principles of implementing successful projects. Various aspects are considered including Costing, budgeting, risk management, human resource change management and scheduling as critical factors to the success of projects.

Course Objectives

The Course aims to provide students with understanding and skills on:

- i) Project Planning, scheduling and Control
- ii) Time, Cost and Quality management
- iii) Building and leading multi-disciplinary and effective teams
- iv) Risk and change management
- v) project monitoring using appropriate project management software

Learning Outcomes

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

- i) Explain project management concepts and roles through all projects
- ii) Manage a project to its satisfactory conclusion on time and within budget
- iii) Integrate the concepts of risk and change management in project management
- iv) Demonstrate the ability to use project monitoring tools and techniques to ensure successful project implementation

Course Outline

The major areas that will be covered include:

- Overview of Project Genesis 5 hours
- Project Planning, scheduling and control 5 hours
- Estimating, Cost and Budgeting 5 hours
- Project Risk Analysis and Management 5 hours
- Project stakeholder analysis and Project Team Structuring 6 hours
- Monitoring Progress – Processes tools and Techniques 5 hours
- Configuration Management 5 hours
- Quality management 4 hours

- Practicals

5 hours

Mode of Delivery

Lecturers and Tutorials

Course Assessment

- | | |
|------------------|-----|
| i) Assignments | 20% |
| ii) Test | 20% |
| iii) Examination | 60% |

25. Research Skills

Course Name	RESEARCH SKILLS
Course Level	Level 3
Course Credit	3CU
Contact hours	45

Course Description

This course provides students with the theoretical and practical concepts of research. It provides the skills, methods and competences that are necessary to collect, analyse, summarise and present quantitative and qualitative data.

Course Objectives

The aims of the course are to:

- Provide students with knowledge of conducting research that achieves pre-determined objectives.
- Describe the various data analysis and reporting techniques.

Learning Outcomes

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

- Demonstrate knowledge of how to conduct research that achieves pre-determined objectives.
- Use the various data analysis and reporting techniques.

Course Outline

- | | |
|--|-----------------|
| • Introduction to research methods | 3 hours |
| • Research Approaches | 3 hours |
| • Problem formulation | 3 hours |
| • Data collection | 6 hours |
| • Data Analysis | 9 hours |
| • Data presentation and report writing | 3 hours |
| • Research ethical issues | 3 hours |
| • Tutorials | 15 hours |

Mode of Delivery

Lectures and Tutorials

Course Assessment

- Tests 20%

- Assignments 20%
- Final examination 60%

24. Embedded Systems Software

Course Name	EMBEDDED SYSTEMS SOFTWARE
Course Level	Level 3
Course Credit	4CU
Contact hours	60 Hours

Course Description

This course presents assembly language programming as the bridge between high-level programming languages and embedded systems hardware with particular reference to the embedded processor.

Course Objectives

The course aims at:

- i) Introducing students to the role of assembly language programming
- ii) Equipping students with skills required to apply a modular approach to assembly language programming with code reuse
- iii) Provide students with knowledge of the instruction set of a typical embedded processor
- iv) Imparting an understanding of memory addressing, the various addressing modes, hardware interrupts and how to use them
- v) Enabling students to integrate assembly language subroutines into a high-level language program

Learning Outcomes

Upon successful completion of this course, the student will:

- i) Understand the role of assembly language programming
- ii) Understand the instruction set of a typical embedded processor
- iii) Be able to employ a modular approach to assembly language programming with code reuse
- iv) Be able to use embedded systems development tools
- v) Understand memory addressing and use various addressing modes
- vi) Understand hardware interrupts and be able to use them
- vii) Be able to integrate assembly language subroutines into a high-level language program

Course Outline

Topics covered include:

- | | |
|---|----------|
| • An introduction to assembly language | 3 hours |
| • The addressing modes, register file, and instruction set of a microcontroller | 12 hours |
| • Timers, handshaking, input and output, and analog to digital conversion | 12 hours |
| • Interrupts | 8 hours |
| • Software control of hardware | 10 hours |
| • Practical lessons (30 hours) | 15 hours |

Mode of Delivery

Lectures and Practicals

Course Assessment

- | | |
|---------------|-----|
| • Tests | 20% |
| • Practicals | 20% |
| • Examination | 60% |

25. Mobile Computing And Applications

Course Name	MOBILE COMPUTING AND APPLICATIONS
Course Level	Level 3
Course Credit	4 CU
Contact hours	60 Hours

Course Description

This course introduces hardware and software concepts of mobile computing. It then discusses mobile application development technologies.

Course Objectives

The aims of this course are:

- i) To introduce students to hardware and software components of mobile computing
- ii) To equip students with an understanding of mobile operating systems as well as mobile multimedia messaging and web services
- iii) To impart knowledge and skills required to design and build professional mobile applications

Learning Outcomes

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

- i) Explain the fundamentals of mobile computing including both the hardware and software components
- ii) Understand the mobile IP stack and mobile web access, technologies and services.
- iii) Demonstrate knowledge of mobile software architecture, operating system platforms and software development tools for mobile software, content and service development.

Course Outline

The topics to be covered include:

- | | |
|--|----------|
| i) Fundamentals of Mobile Terminal Hardware | 2 hours |
| ii) Radio, DSP, Memory and CPU components | 3 hours |
| iii) Base Station side radio interface standards | 2 hours |
| iv) The Mobile IP stack and mobile web-browsing | 2 hours |
| v) The WAP-protocol & Location Information | 3 hours |
| vi) Principles of Multimedia Messaging (SMS, MMS) and web services | 4 hours |
| vii) Introduction to Mobile Operating System Platforms (OSPs) | 6 hours |
| viii) Mobile Applications User Interface | 2 hours |
| ix) Software set-up in modern Mobile Terminals | 6 hours |
| x) Fundamentals of Mobile Content | 6 hours |
| xi) Mobile web content design | 8 hours |
| xii) The .mobi top level domain (TLD) initiative | 1 hour |
| xiii) Practical Lessons (30 hours) | 15 hours |

Mode of Delivery

Lectures and Practicals

Course Assessment

- Tests 20%

- Practical (Project) 20%
- Examination 60%

26. Software Engineering Mini Project

Course Name	SOFTWARE ENGINEERING MINI PROJECT
Course Level	Level 3
Course Credit	4CU
Contact hours	60

Course Description

The course introduces students to the integrated team approach to undertaking software Engineering projects from the point of requirements analysis to deploying a tested and well engineered solution, with the assistance of a Project Supervisor (s). It also plays a crucial role in exposing students to building highly effective teams which is very important in “real life” industrial experience.

Course Objectives

The course aims to provide students with understanding and skills in:

- i) Project management concepts and roles
- ii) Planning and controlling projects
- iii) Considerations for commercialization of software products
- iv) Software Requirements Analysis, specification and Design
- v) Choosing appropriate Software Engineering Processes, Tools and Methods
- vi) Professional Software construction
- vii) Software maintenance and Management
- viii) Software documentation
- ix) Professional Software quality assurance and Testing

Learning Outcomes

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

- Explain the nature of Software Engineering projects and their management
- Demonstrate understanding of Finance Planning, Budgeting and Costing in Software Engineering Projects
- Integrate the concepts of Risk Management, Human resource management, control of time and deadlines through application of tools and techniques
- Demonstrate the ability to develop a business plan as part of the Project Proposal for their choice of research project.
- Demonstrate creativity and innovation in the developing software solutions
- v) Develop the requirements and design specifications
- vi) Attest and practice high-quality Project Management skills in enhancing individual effectiveness and productivity
- vii) Develop Software using state of the art technologies
- viii) Document the software in accordance to well known practices

Course Outline

The major areas that will be covered include:

- | | |
|--|----------|
| ○ Project Proposal and Business Plans | 6 Hours. |
| ○ Requirements Analysis, Specifications and Design | 8 Hours. |
| ○ Project Management | 4 Hours |
| ○ Software Methodology, Tools and Techniques | 6 Hours |
| ○ Software Construction | 20 Hours |
| ○ Software Quality Assurance and Testing | 10 Hours |
| ○ Project Report Write Up | 6 Hours |

Mode of Delivery

Presentations, Project supervision

Course Assessment

- | | |
|---------------------------|-----|
| i) Project Proposal | 20% |
| ii) Project Demonstration | 20% |
| iii) Project Report | 30% |
| Presentation | 30% |

27. Industrial Training

Course Name	INDUSTRIAL TRAINING
Course Level	Level 3
Course Credit	4CU
Contact hours	60

Course Description

Industrial Placement promotes and develops long-term strategic teaching and research alliances between higher education institutions and industry. It plays a crucial role in promoting and facilitating practical training by providing students the opportunity to apply advanced knowledge and skills acquired in class to real world industry based problems and also gain exposure to “real life work experience.”

Course Objectives

The course aims to enable students:

- Understand the different types of Computing careers and their roles in a real world business organization.
- Apply and translate computing knowledge and best practice to real industry based problems
- Acquire soft skills to enhance effective participation in group based industry projects
- Learn and practice good working ethics and communication skills with industry based staff and Supervisors.

Learning Outcomes

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

- i) Explain the different types of Computing careers and their roles in a real world business organization.
- ii) Demonstrate ability to apply and translate computing knowledge and best practice to industry based problems
- iii) Demonstrate creativity and innovation in solving problems related to real-life projects
- iv) Demonstrate soft skills for enhanced effective participation in group based industry projects
- v) Demonstrate and practice good working ethics, communicate effectively and exhibit good interpersonal skills with industry based staff and Supervisors.

Course Outline

The major areas that will be emphasized during industrial training include

- | | |
|--|----------|
| i) Orientation to Industry, Career Prospects & Human Resource Policy | 4hours |
| ii) ICT Infrastructure in Industry | 6hours |
| iii) Communication Network Infrastructure | 4hours |
| iv) Information Management | 10hours |
| v) Systems Administration | 4hours |
| vi) Website Development | 4hours |
| vii) IT Quality Assurance | 4hours |
| viii) Audit and Review ICT Infrastructure Maintenance Manuals | 4hours |
| ix) Audit and Review IT and Communication Policies | 4hours |
| x) Audit and Review Information Security | 4 hours |
| xi) Project Report Writing | 12 hours |

Mode of Delivery

Mentorship

Course Assessment

- | | |
|--|-----|
| • Evaluation Report by Industry Based Supervisor | 40% |
| • Interview by Academic Supervisor | 30% |
| • Written Industrial Training Project Report | 30% |

1. Real-Time And Embedded Applications

Course Name	REAL-TIME AND EMBEDDED APPLICATIONS
Course Level	Level 4
Course Credit	4CU
Contact hours	60 Hours

Brief Course Description

The course focuses on the process as well as fundamentals of integrating microprocessor-based embedded system elements for digital command and control of typical embedded hardware systems. In this course, students will design and build a microprocessor-based embedded system application requiring integration of sensor/actuator devices, A/D and D/A I/O interfaces, single and multi-core microprocessors, commercial real-time operating system, and multi-tasking application software.

Course Objectives

This course aims at:

- i) Introducing students to the rate monotonic theory and how to apply it to real-time applications
- ii) Equipping students with knowledge of the processes and fundamentals of integrating microprocessor-based embedded system elements to realize applications that meet strict timing and performance requirements
- iii) Imparting practical skills of designing and integrating a real-time operation system with a microprocessor to host real-time service data processing

Learning Outcomes

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

- i) Demonstrate knowledge of rate monotonic theory and how to apply it to real-time applications
- ii) Understand the process and fundamentals of integrating microprocessor-based embedded system elements to realize systems that not only meet functional requirements, but timing and performance requirements as well
- iii) Use practical skills to design and integrate a real-time operation system with a microprocessor to host real-time service data processing

Course Content

- Rate Monotonic Theory 6 hours
- Real-time Operating Systems 8 hours
- Embedded system architectures 6 hours
- Sensor and Actuator IO 9 hours
- Real-time embedded test equipment, tools and performance 4 hours
- Real-time applications 12 hours
- Practical Lessons (30 hours) 15 hours

Mode of Delivery

Lectures and Practicals

Course Assessment

- Tests 20%

- Practicals 20%
- Examination 60%

28. Business Law

Course Name	Business Law
Course Level	Level 3
Course Credit	4 CU
Contact hours	60

Course Description

This course is designed to impart knowledge to students on the various aspects relating to Business undertakings and operations. It covers the Legal aspects of a business like the law of undertaking and operations. It also covers the legal aspects of a business like the Law of Contract, arbitration, Law of Agency, Labor Laws and the Company act in General.

Course Objectives

The Course aims to provide students with understanding and skills on:

- Various aspects relating to undertakings and operations
- Legal principles relating to corporate business and governance

Learning Outcomes

Upon Completion of the course, students shall be able to:

- Explain the various aspects of any one national legal system
- Explain of the various aspects relating to undertakings and operations
- Understand and explain the Legal principles relating to corporate business and governance

Course Content

i)	The Introduction to the Uganda Legal System and Structure of Courts	8Hours
ii)	Evolution of Companies as business associations in Britain	4Hours
iii)	Promotion and Formation of a Company	5 Hours
iv)	The Theory/Concept of Corporate status/Personality	4Hours
v)	Pre-incorporation Contracts	5Hours
vi)	The Ultra-Vires Doctrine	4Hours
vii)	Lifting the Veil of the Incorporation	5Hours
viii)	Management of a Company	4Hours
ix)	Company Meetings and Resolutions	5Hours
x)	Raising the Minimum Capital	5Hours
xi)	The Ministry's members rights	5Hours
xii)	Winding up and Dissolution / Liquidation of a Company	6Hours

Mode of Delivery

Lectures and Tutorials

Assessment

- Course Work 40%
- Final Exam 60%

29. Professional Issues In Computing

Course Name	PROFESSIONAL ISSUES IN COMPUTING
Course Level	Level 4
Course Credit	3 CU
Contact hours	45

Course Description

The course provides awareness of the legal, social, ethical, professional and contemporary issues commensurate with the practice of a computing career.

Course Objectives

The Course aims to provide students with understanding and skills on:

- Standards and professionalism in a Computing Career
- Legal environment within which a computing professional operates
- The professional societies that govern the behavior of computing professionals.
- Professional responsibilities to client, employer and public

Learning Outcomes

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

- i) Appreciate the role of standards for a practicing computing professional
- ii) Practice a computing career with a due regard to legal, ethical and social issues

Course Outline

The major areas to be covered include:

i)	The computing profession and standards	3 hours
ii)	Computing and society	3 hours
iii)	Ethics and the social context of computing	6 hours
iv)	Methods and tools for analysis of ethical arguments	3 hours
v)	Risks and liabilities of safety-critical systems	3 hours
vi)	Intellectual property rights	2 hours
vii)	Privacy and civil liberties	2 hours
viii)	Social implications of the Internet	3 hours
ix)	Computer Misuse	5 hours
x)	ICT legal framework	5 hours
xi)	Tutorials and group work (20 hours)	10 hours

Mode of Delivery

Lectures and Tutorials

Course Assessment

- Group Assignment 20%
- Test 20%
- Examination 60%

30. Software Design Patterns

Course Name	SOFTWARE DESIGN PATTERNS
Course Level	Level 4
Course Credit	3 CU
Contact hours	45

Course Description

This course explores advanced principles of object-oriented design by studying key software design patterns. The patterns are drawn from a variety of sources and illustrated through examples and case studies. These are presented in an object oriented programming language e.g. Java, C++ or C sharp.

Course Objectives

The Course aims to provide students with understanding and skills in:

- Designing effective and maintainable complex software systems of high quality by reusing/applying design patterns
- Evaluating a design, identifying common problems, and how to fix these problems through refactoring

Learning Outcomes

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

- Describe the purpose of design patterns.
- Describe a range of software patterns giving examples of their use
- Understand the ways that design patterns are documented and classified.
- Select and apply design patterns to practical software design problems
- Select and apply design patterns to realistic business situations through a series of hands-on exercises

Course Outline

The major areas that will be covered include:

- | | |
|---|--------|
| i) Introduction to design patterns | 3Hours |
| ii) Creational, structural and Behavioral Patterns | 6Hours |
| iii) Relationship between Patterns | 3Hours |
| iv) Patterns for concurrent and distributed systems | |
| 3Hours | |
| v) Advanced Patterns | 3Hours |
| vi) Automation of Patterns and Anti-patterns | |
| 3Hours | |
| vii) Introduction to Aspect-Oriented design | 3Hours |
| viii) Introduction to business process modeling | |
| 3Hours | |
| ix) Patterns for coding, testing and concurrency control. | 3hours |

x) Practical and Case Studies of Applications of Design Patterns
15Hours

Mode of Delivery

Lectures, Tutorials, Practical's

Course Assessment

- Practicals 20%
- Test 20%
- Examination 60%

31. Internet Programming

COURSE NAME	INTERNET PROGRAMMING
COURSE LEVEL	LEVEL 4
COURSE CREDIT	3CU
CONTACT HOURS	45 HOURS

Course Description

This course covers ASP.NET development fundamentals. It provides hands on training of server-side technologies, C# (C Sharp) programming language used for ASP.NET and database development utilizing Microsoft SQL Server Desktop Engine (MSDE). Techniques for integrating ASP.NET pages with the advanced client-side (DHTML) technologies are covered.

Course Objectives:

The aims of the course are:

- To introduce students to the .Net development environment
- To enable students to apply core ASP.NET technologies to develop Web applications
- To train students to author server-side ASP.NET code in C#
- To train students to create Web applications using ADO.NET to interact with SQL Server
- To enable students to write ASP.NET pages that integrate into traditional DHTML-driven pages.

Learning Outcomes

On successfully completing this course, students shall be able to:

- Set up a .Net development including environment, including MSDE
- Apply core ASP.NET technologies to develop Web applications
- Write server-side ASP.NET code in C#
- Create web applications using ADO.NET to interact with SQL Server
- Write ASP.NET pages that integrate into traditional DHTML-driven pages

Course Outline

This course covers:

i) ASP.NET development fundamentals	5hours
ii) Server-side technologies	5hours
iii) C# (C Sharp) programming language commonly used for ASP.NET	5hours
iv) Database development with MSDE	5hours
v) integrating ASP.NET pages with client-side DHTML technologies	5hours
vi) Practical Lessons (40 hours of Practicals)	20hours

Mode of Delivery

Lectures, Case studies, Tutorials, Practicals, Group work

Course Assessment

• Tests	20%
i) Practicals	20%
ii) Examination	60%

32. Computer Graphics

COURSE NAME	COMPUTER GRAPHICS
COURSE LEVEL	LEVEL 4
COURSE CREDIT	3CU
CONTACT HOURS	45 HOURS

Course Description

This course introduces students to the basic principles of computer graphics. Students are exposed to the knowledge and skills of using computer graphics to portray real or imagined concepts.

Course Objectives

The course aims to:

1. Introduce students to computer graphics systems and concepts.
2. Provide students with an in-depth knowledge of the creation and analysis of computer graphics objects.

Learning Outcomes

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

- i) Explain computer graphics systems and concepts.
- ii) Demonstrate the ability to create and analyze computer graphics objects.

Course Content

The topics to be covered in this course include:

• Computer graphics principles, terminologies and applications	3 hours
• Graphics hardware	3 hours
• Graphics design	3 hours
• 2D and 3D transformations	6 hours
• Graphics algorithms	5 hours
• Simulating graphics	4 hours
• Graphics visualisation	3 hours
• Graphics standards	3 hours
• Practical Lessons (40 hours)	20 hours

Mode of Delivery

Lectures, practicals, discussions, demonstrations

Course Assessment

i) Tests	20%
ii) Practical	20%
iii) Examination	60%

33. Software Evolution

COURSE NAME	SOFTWARE EVOLUTION
COURSE LEVEL	LEVEL 4
COURSE CREDIT	3CU
CONTACT HOURS	45 HOURS

Course description

This course shows how Software evolution helps to implement and revalidate possible major changes to a system without the need to predict how user requirements will change. An existing large system is never complete and continues to evolve. As it evolves, the complexity of the system grows and Software evolution ensures continuous reliability and flexibility of the system.

Course Objectives

The aims of this course are to teach students:

- The concepts and importance of software systems evolution
- The software evolution process
- Models and Laws of software systems evolution
- The causes and impact of software systems evolution
- The properties of software systems evolution
- How selected software systems are analyzed based on properties of their evolution
- How to interpret the implications of software systems evolution.

Learning Outcomes

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

- Explain the concepts of software systems evolution
- Demonstrate the understanding of the causes and impact of software systems evolution
- Explain the laws of software systems evolution
- Describe the properties of software systems evolution
- Analyze selected software systems to understand properties of their evolution
- Interpret the implication of Evolution for software systems.

Course Content

To achieve the objectives of this course, the topics covered include:

- | | |
|---|---------|
| i) Separate compilation | 3hours |
| ii) Design issues and common design patterns | 3 hours |
| iii) Verification and validation | 3 hours |
| iv) Integrating components | 3 hours |
| v) Documentation | 2 hours |
| vi) Issues in object-oriented programming | 3 hours |
| vii) Parallelism | 3 hours |
| viii) event-centered programming | 3 hours |
| ix) software reuse | 3 hours |
| x) The Models and Laws of Software Evolution | 4 hours |
| xi) The forward and Reverse engineering and reengineering tools | 6 hours |
| xii) Legacy system evolution | 3 hours |
| xiii) Tutorials (12 hours) | 6 hours |

Mode of Delivery

Lectures and Tutorials

Course Assessment

- Tests 20%
- Assignment 20%
- Examination 60%

34. Entrepreneurship And Business

Course Name	ENTREPRENEURSHIP AND BUSINESS
Course Level	Level 1
Course Credit	3CU
Contact hours	45 Hours

Course Description

The course introduces the students to the basic concepts in entrepreneurship, identification of business opportunities, business evaluation and analysis. It provides students with the skills needed to effectively identify, organize, develop, and manage own business ventures.

Course Objectives

The course aims to provide students with:

- Knowledge about the origins of entrepreneurship and an entrepreneur
- Knowledge about entrepreneurship development
- Insight on how to Identify, evaluate, and select business opportunities
- Skills to perform self-evaluation
- Knowledge on how to carry out feasibility and viability of an investment opportunity
- Insight to exploit the Entrepreneurial Environment provided by the political, socioeconomic and technological conditions

Learning Outcomes

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

- i) Explain the origins of entrepreneurship and describe an entrepreneur
- ii) Outline and explain the stages of Entrepreneurship Development
- iii) Explain the barriers to Entrepreneurship
- iv) Perform self evaluation to match business opportunities
- v) Analyze the entrepreneurial environment
- vi) ensure start-up, survival and sustainability of an investment opportunity
- vii) Identify their own personal entrepreneurial potential, and competences
- viii) Identify, and exploit business opportunities and resources

Course Outline

i) The concepts of entrepreneurship	3 hours
ii) Theories of Entrepreneurship	4 hours
iii) Entrepreneurial Characteristics	3 hours
iv) Entrepreneurial Process	3 hours
v) The Entrepreneurial Environment	3 hours
vi) Entrepreneurship Development	5 hours
vii) Business start-up and growth	6 hours
viii) Financial basics	3 hours
ix) Building Entrepreneurial communities	3 hours
x) Management of Entrepreneurial Risk	3 hours
xi) Entrepreneurial ventures	3 hours
xii) Tutorials (12 hours)	6 hours

Mode of Delivery

Lectures and Tutorials

Course Assessment:

- Tests 20%
- Assignments 20%
- Examination 60%

35. Information Security

Course Name	INFORMATION SECURITY
Course Level	Level 4
Course Credit	4CU
Contact hours	60 Hours

Course Description

This course familiarizes the students with existing security problems and the techniques that can be applied to overcome them. Focus is given to security of information that is stored in files or in databases.

Course Objectives

The aims of this course are:

12. To introduce students to contemporary threats to information security
13. To provide students with knowledge about contemporary solutions (tools and techniques) to information security
14. To equip students with skills required to prevent, identify and mitigate the threats to information security

Learning Outcomes

Upon completion of this course, the students shall be able to:

- 13 Identify contemporary threats to information security
- 14 Apply contemporary solutions to the threats of information security using appropriate tools and techniques
- 15 Propose solution of prevent, identify and mitigate the threats to information security

Course Outline

- | | |
|---|----------|
| • Data security concepts and examples | 3 hours |
| • Information security threats | 6 hours |
| • Information security controls | 6 hours |
| • Information security Tools and Techniques | 8 hours |
| • Information security Risk analysis | 6 hours |
| • Information security and social aspects | 6 hours |
| • Practicals (20 hours) | 10 hours |

Mode of Delivery

Lectures and practicals

Course Assessment

- | | |
|----------------------|-----|
| 13 Tests | 20% |
| 14 Assignments | 20% |
| 15 Final examination | 60% |

36. Final Year Project

Course Name	FINAL YEAR PROJECT
Course Level	Level 4
Course Credit	4CU
Contact hours	60

Course Description

This course provides the student with the opportunity to independently undertake a research project from the point of requirements analysis to deploying a tested and well engineered solution, with the assistance of a Project Supervisor(s). The student has to demonstrate a high level of professionalism and best practice in terms of documentation and software engineering standards.

Course Objectives

The Course aims to provide students with understanding and skills on:

- Developing of a concept and plan for a commercially viable software project
- Determining requirements and coming up design specification
- Choosing appropriate Software Engineering Processes, Tools and Methods
- Professional Software construction and deployment
- Software maintenance and Management
- Professional Software quality assurance and Testing
- Software project documentation

Learning Outcomes

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

- Explain the nature of Software Engineering projects and their management
- Demonstrate understanding of Finance Planning, Budgeting and Costing in Software Engineering Projects
- Integrate the concepts of Risk Management, Human resource management, control of time and deadlines through application of tools and techniques
- Demonstrate the ability to develop a business plan as part of the Project Proposal for their choice of research project.
- Demonstrate creativity and innovation in the developing software solutions
- ix) Develop the requirements and design specifications
- x) Attest and practice high-quality Project Management skills in enhancing individual effectiveness and productivity
- xi) Develop Software using state of the art technologies
- xii) Document the software in accordance to well known practices

Course Outline

The major areas that will be covered include:

- | | |
|--|---------|
| ○ Project Proposal and Business Plans | 6 Hours |
| ○ Requirements Analysis, Specifications and Design | 8 Hours |
| ○ Project Management | 4 Hours |
| ○ Software Methodology, Tools and Techniques | 6 Hours |

- | | |
|--|----------|
| ○ Software Construction | 20 Hours |
| ○ Software Quality Assurance and Testing | 10 Hours |
| ○ Project Report Write Up | 6 Hours |

Mode of Delivery

Presentations and Project supervision

Course Assessment

- | | |
|---------------------------|-----|
| i) Project Proposal | 20% |
| ii) Project Demonstration | 20% |
| iii) Project Report | 30% |

APPENDIX 1: DETAILS OF THE COURSE NAMES OF THE SOFTWARE ENGINEERING PROGRAMME FROM MAKERERE UNIVERSITY IN COMPARISON WITH THE SOFTWARE ENGINEERING THEMES OUTLINED IN SWEBOK