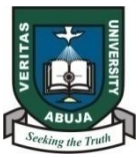
VERITAS UNIVERSITY, ABUJA



DEPARTMENT OF SOFTWARE ENGINEERING
(The Centre of Excellence)

UNDERGRADUATE HANDBOOK

2021 - 2026

From the Desk of the Head of Department

You are welcome to the Department of Software Engineering (DSE), Veritas University, Abuja, "The Centre of Excellence".

Here in SE department, we aim to:

- ✓ Deliver the best software engineering theoretical foundation to all our students
- ✓ Provide a rich practical software engineering education for practical problem solving.
- ✓ Offer a project-based software engineering education for skill acquisition and growth.
- ✓ Give a well-grounded laboratory-based software engineering education to promote and instill hands-on practice and culture.
- ✓ Provide a software engineering education with quality industrial exposure via internships
- ✓ Create in students the awareness of and enthusiasm for computing and its capacities.
- ✓ Involve students in an intellectually stimulating and satisfying experience of learning and studying.
- ✓ Provide a broad and balanced foundation in software engineering knowledge and practical skills.
- ✓ Develop in students through an education in software engineering a range of transferable applicable skills in computing that touches all aspects of human endeavors.
- ✓ Generate in students an appreciation of the importance and relevance of software engineering in industrial, economic, technological and social contexts.
- ✓ Train students in providing solutions to meet societal computing needs.
- ✓ Rigorously prepare students for today's industry and the challenges it brings.
- ✓ Create in students the energy to be curious and motivated about technology and to be ambitious to make a difference in the tech industry.
- ✓ Challenge students to constantly challenge the status quo in technology, to think out of the box, to innovate and create new technologies for the common good of society.
- ✓ Adequately get students ready for the job market on graduation.
- ✓ Prepare students to be entrepreneurs with suitable startups with cutting edge skills.
- ✓ Essentially, in SE department, we bridge the gap between theory and practice.

We encourage you all to be resolute to learn. Our staff members are committed to serve you and to ensure that your dreams come true.

This programme is a fully practical one and has intensive lab sessions. Students should therefore avail themselves of the opportunities the course provides and be prepared to work hard and be committed to excellence, both in theory and practice.

Dr Emmanuel Mkpojiogu
Head of Department, Software Engineering

TABLE OF CONTENTS

1. ABOUT THE DEPARTMENT

1.1.Mission of the University

To provide its students with an integral and holistic formation that combines academic and professional training with physical, moral, spiritual, social and cultural formation together with formation of Christian religious principles and social teachings of the Catholic Church.

1.2 Philosophy of Software Engineering Programme

Nigeria (and indeed sub-Saharan Africa) represents virgin territory for the software industry and the field in turn presents huge opportunities for the region within the contest of an expanding global economy. It is well known that the software industry is the only one that wealth can be created from zero or near-zero capital — only intellectual capital is needed. There is therefore the need for Nigeria to grow its own crop of software engineers as a force for sustainable socio-economic development.

The Philosophy and Mission Statement underlying the Software Engineering programmes is aimed at achieving the goals and objectives of the National Policy on Industrialisation and Self-Reliance. This is to be achieved through:

(i) Broad—based foundation in Computing, Science and Engineering and Technology as well as specialized knowledge and practice in Software Engineering.

(ii) Practical exposure to application of Software Engineering to problem solving

(iii) Adequate training in human and organisational behaviour and management in the software development life-cycle.

(iv) Developing in the programme, entrepreneurial knowledge, a sense of public responsibility and a spirit of self-reliance.

(v) Nurturing of partnership between the institution and the software industry effective programme delivery.

(vi) Creating an awareness and understanding of the moral, ethical, legal. And professional obligations needed to function as part of the computing a global society.

(vii) Creating an awareness and understanding of the need to develop leadership and team building skills to maximize the benefits of Software Engineering education and its application to solving problems.

The general philosophy therefore is to produce graduates with high academic and ethical standards and adequate practical exposure for self-employment as well as being of immediate value to the software industry and the community in general.

1.3.Aims/Objectives

The general goal and objectives of Software Engineering education and training should be in consonance with the realisation of national needs and aspirations vis-å-vis industrial development and technological emancipation. The graduates must therefore be resourceful, creative, knowledgeable and able to perform the following functions:

i. To appreciate the importance of computer science (as a base) in suchareas as principles of programming, algorithm, data structures, databases and programming languages.
ii. To develop and utilize the practical skills acquired in software architecture and design, software metrics, verification and validation, requirements and analysis and the software engineering process for the production of software-based systems.
iii. To develop expertise in programming in a number of different languages with emphasis on the production of robust, reliable, cost-effective and secure codes that are based on sound design and development principles and adapted to the needs of a developing and emerging economy such as Nigeria in a technically professional manner.
iv. To be able to exercise original thought, have good professional judgment and be able to take responsibility for the execution of important tasks as programmers, system analysts, software developers, web developers, software consultants, system administrators, IT project managers, system engineers and entrepreneurs, etc.
v. To be able to produce and manage high-quality software-based solutions with long life-cycles especially for large or complex systems.
vi. To develop the understanding and engineering and entrepreneurial skills needed to become the architects and project leaders building system in which software plays a critical vole
vii. To leverage on Software Engineering as the driving force behind the new technologies that are transforming the way we live and work.
viii. To have the requisite knowledge and skill base for further academic and professional development in Software Engineering

The Software Engineering programme will focus on imparting the knowledge and practical skills to enable students understand the principles and practice of software systems design, development and maintenance. This should enable them to be able to balance software system design and development with safety, reliability, cost and scheduling especially for large scale systems.

1.4.Learning Outcome:

1.4.1 Regime of Subject Knowledge

- Have a thorough knowledge of computing, scientific and engineering practice an, theory in software engineering and be able to extend this knowledge through self-le study.
- Understand the impact of globalization on computing and software engineering
- Recognize the challenges and opportunities for the software industry in Nigeria and the region.
- Identify requirements for specialised computing systems and proffer solutions them.
- Use and, where appropriate, modify for specific use, established system development methods.
- Explain the relationships between computer systems and other natural and artificial systems in the modern oriel at appropriate levels or abstraction.
- Explain the concepts of computer programming and critically evaluate and predict their utility in models, tools and applications.

- Demonstrate advanced, specialist theoretical and practical knowledge in a range of computer science sub-fields.
- Explain legal issues relating to computing: intellectual property, data protection, computer misuse and health and safety.
- Explain the principles and practice of software engineering in a modem industrialcontext

1.4.2. Competences and Skills

- Analyse and abstract problems and propose and apply effective solutions.
- Apply software engineering theory, principles, tools and processes, as well as the theory and principles of computer science and mathematics, to the development and maintenance of complex, scalable software systems.
- Participate productively on software project teams involving others from a variety of disciplines and backgrounds.
- Develop and critically evaluate specifications for specialist computer system
- Elicit, analyse and specify software requirements through a productive working relationship with project stakeholders.
- Apply software engineering best practice to the development or computer systems
- Demonstrate software engineering application domain knowledge having engineered a software product of value to a specific organization or to address a societal need.
- Implement specialist computer systems from given specifications.
- Plan and manage large scale projects.
- Communicate requirements and proposals for computer systemsother computing professionals.
- Communicate effectively through oral and written reports, and software documentation.
- Evaluate the impact of potential solutions to software engineering problems in a global society, using their knowledge of contemporary issues.
- Employ an entrepreneurial approach to provide sustainable software engineering solutions.
- Identify the common needs of industry from computer systems and apply controlled comprise in meeting requirements.
- Design and execute methodologically sound computing, scientific and engineering studies.
- Demonstrate advanced specialist skills in addressing the problems of Computer Science and its sub-fields.
- Understand, evaluate, synthesize and apply complex ideas.

1.4.3. Behavioral Attributes

- Apply appropriate codes of ethics and professional conduct to the solution of software engineering problems.
- Explain the issues of professionalism in computing including the need for continuing professional development.
- Assess the nature of intellectual property and its ownership, and respect it accordingly.
- Identify resources for determining legal and ethical practices in other countries as they apply to computing and software engineering.

1.5. Rationale/Justification

Compatibility of programme with institutional mission:

In line with and within the bounds of the mission of Veritas University, the Department shall strive towards sustainable human and technological development through the utilization of the resources available within its environs in teaching, research and community service while maintaining a good conducive and friendly environment for life.

Relevance to national needs (evidence of relevance from National Manpower Board to be attached).

In meeting the manpower needs for national development, the programme will:

a. Enable the graduates develop the ability to apply acquired knowledge and skills to solving theoretical and practical problems in computing and software engineering.
b. Enable the graduates develop relevant software engineering skills for national development and societal needs,
c. Enable the graduates train others to acquire knowledge and skills necessary to function effectively in a computing-driven environment.

1.6. Admission Requirement

State clearly the admission requirements for the proposed programme. It must meet the general minimum admission requirement for degree programmes.

Candidates can be admitted by one of the following three ways:

* The Unified Tertiary Matriculation Examination (UTME)
* Direct Entry
* Inter-University Transfer

1.6.1 UTME Entry Mode

The minimum academic requirement is credit level passes in five subjects at O'Level in nationally recognized examination including English Language, Mathematics, Physics and two other science subjects which are relevant to the programme at not more than two sittings. In addition, an acceptable pass in the Unified Tertiary Matriculation Examination (UTME) with relevant subject combinations is also required for admission into 100 Level.

1.6.2 Direct Entry Mode

Candidates seeking admission into the programme through Direct Entry must have passes at GCE 'A' Level/IJMB or its equivalent in a minimum of two Science subjects relevant to the programme to be considered for admission into the 200 Level. This is in addition to fulfilling the requirement of a minimum of credit level passes in five relevant subjects at 'O' Level as indicated above.

1.6.3 Inter- and Intra-University Transfer Mode

Students can transfer into 200-Level BSc Software Engineering Programme provided they have a minimum of 1.5 CGPA at the end of their 100 Level in their previous programmes or universities. This is in addition to the UTME requirement.

1.7 Academic Staff

Table 1. List of Existing Academic Staff for the Programme

S/N	Name of Academic Staff	Area of Specialisation	Discipline	Qualification	Rank	Status
1.	Dr Emmanuel Mkpojiogu	Software Engineering	Computing	BSc; MSc; PhD	Senior. Lecturer & HOD	Full-time
2.	Dr Deborah Ebem	Software Engineering	Computing	BSc; MSc; PhD	Associate Professor	Full-time
2.	Mr. Ikechukwu Izegbu	Software Engineering	Computing	BSc; MSc	Lecturer II	Full-time
3.	Mr. Dako Apaleohkai Dickson	Software Engineering	Computing	BTech; MTech	Assistant Lecturer	Full-time
4.	Ms Mary Kama	Software Engineering	Computing	BSc; MSc	Assistant Lecturer	Full-time
5.	Mr. Usoh Ikouwen Ufort	Software Engineering	Computing	BSc; MSc	Assistant Lecturer	Full-time
6.	Mrs. Immaculate Agubata	Software Engineering	Computing	BSc; MSc	Assistant Lecturer	Full-time
7.	Mr. Oladayo Tosin Akinwande	Software Engineering	Computing	BTech; MTech	Assistant Lecturer	Full-time
8.	Mr. Supreme A. Okoh	Software Engineering	Computing	BEng; MEng	Assistant Lecturer	Full-time

Table 2. List of Existing Non-Teaching Staff for the Programme

S/N	Name of Staff	Area of Specialisation	Discipline	Qualification	Rank	Status
1.	Mr. Christian Nwanagba	Software Engineering	Computing	BSc	Laboratory Technologist I	Full-time
2.	Mr. Pius Ogah Agba	Software Engineering	Computing	BSc	Laboratory Technologist II	Full-time
3.	Miss Nelly Ibe Agwu	Software Engineering	Computing	HND	Laboratory Technologist II	Full-time
4.	Mr. Adrian Sallama Suga	Secretary	English/PES	NCE	Secretary	Full-time

2. Registration

2.1. General

All new students, on arrival at the University, should proceed for registration by contacting the under listed for the following:

i. Faculty Officer - for screening, clearance and course registration forms
ii. Academic Officers - to collect registration files and forms and to have them signed
iii. Bursary (Students) - to pay all necessary accommodation fees
iv. Student Affairs Officer - for hostel
v. University Library - for registration

vi. University Health Centre - for registration and medical examination
vii. Department - for Departmental registration/document.

2.2 Course Registration

At the beginning of every semester, all students are required to follow procedures prescribed by the Examination and Records Unit of the Registry and register manually and on-line. A student is not fully registered for an academic session unless the manual and on-line procedures have been completed within a prescribed period. A student who registers late shall pay a late registration fee to be determined from time to time by the University.

2.2.1 Late Course Registration

A student, who fails to register and submit course registration form within the prescribed registration period for the semester, shall pay a late registration fee to be determined by the University. Failure to register and submit the said course registration form within one month of resumption attracts a higher late registration fee to be determined by the University from time to time.

Students who attend lectures in courses that they have not registered for shall do so only with the express permission of the course lecturer(s). However, such students shall not earn any credit units from participation in such courses.

2.2.2 Deferment of Admission/Studies

A candidate offered admission by the University shall be allowed to defer the admission if the following conditions are satisfied:

i. Accept the offer of admission by completing the University acceptance form and online registration
ii. Pay all prescribed fees for the session in which student was admitted
iii. Apply to the Registrar in writing for the deferment of admission or studies to a specific academic session
iv. Apply in writing to the Registrar for resumption on expiration of the deferment period.

The maximum period for deferment of admission/studies shall be one academic session unless otherwise extended by approval of Senate. A student who fails to return to the University at the expiration of the deferment period shall forfeit the right of admission/studies

2.2.3 Further Condition for Voluntary Withdrawal

Any student who fails to register and attend lectures for one academic year without approval of Senate shall be considered to have voluntarily withdrawn from the programme.

3. Work Load

3.1 Full-time and Residency Requirements

All undergraduate programmes offered in the Department are full-time and fully residential. Students may not undertake any regular paid employment during the course of the programme.

A student shall normally register for a minimum of thirty (30) and a maximum of forty- eight (48) credit units in any academic year, except in the case of direct entry and inter-university transfer students. This means that a student should take between 15 and 24 credits in any given semester and no student can earn more than forty-eight (48) credit units at the end of an academic year. In the special case of direct entry and inter-university transfer students, who follow a three-year degree programme, this maximum may be exceeded by the number of credit units assigned to the General Studies courses, VUA Theology courses, and Students' Industrial Work Experience Scheme (SIWES) which they must take.

3.2 Audited Courses

A student may register to audit a course unit outside his/her programme of study according to his/her interest, but subject to departmental approval. A student may write examinations in such audited courses, and the scores for such examinations reflected in his/her academic records, but he/she may not earn credits for them.

3.3 Conditions for Probation

If at the end of the session, a student's Cumulative Grade Point Average (CGPA) is less than 1.50, then he/she will be placed on probation for a specified period of one full session. A student who is on probation should re-register for all the failed courses before registering the current ones. Such a student should not exceed a credit load of 15 for that semester. If the student still has a CGPA of less than 1.50 at the end of the session, he/she is advised to withdraw from the programme. He/she may shop for admission in another programme.

3.4 Conditions for Withdrawal

If at the end of a probation period, a student's CGPA is still less than 1.50, the student will be asked to withdraw from his/her programme of study. A student who is so withdrawn need not leave the University; rather, he/she may transfer to another programme in any Department within and outside the Department that may be willing to accept him/her.

4. The Course Credit System

Teaching in the Department is by the 'course system'. This is referred to as a "quantitative" system or organization into unit courses which are examinable, and for which students earn credit(s) if passed. The courses are arranged in a progressive order of levels of academic progress, e.g., Level or Year 1 courses are 100 level: 101, 103, 105, 102, 104, 106. Level 2 or year 2 courses are 200 level, e.g., 211, 213, 215, 217 etc. The numbering of courses enables students to immediately know those courses offered during the first or the second semester as well as courses taken in semesters as well as courses taken in various years.

4.1 Coding System

Three-digit coding system has been adopted for all the courses in the University as follows:

First digit represents the **Level of studies; Third** is the **Semester.**

5.Course Classification

Courses are classified as follows:

Compulsory Courses: These are the core courses within the discipline that must be taken and passed at a grade not below 'D'

Elective Courses: These are optional courses within the department for the purpose of fulfilling the minimum requirements for the award of degree.

General Studies Courses: These are University-wide courses that must be offered and passed by every undergraduate student in the University irrespective of discipline. These are General Studies (GES) and Theology (THG) courses.

6. Examination

6.1Setting of Questions

All examinations should have:

i. First Examiners (normally the Academic Staff Member in charge of the course) and

ii. Second Internal Examiner (a Moderator) as may be approved by the Departmental Board of examiners.

Questions to be set and the number to be answered shall be at the discretion of the examiner(s) subject to the approval of the Departmental Board of Examiners usually consisting of all the academic staff of the Department. In the case of 400 level examinations, External Examiners shall participate in the moderation of question papers and vetting of the answer scripts. Examiners shall be expected to submit question papers, marking guide, marked answer booklets and raw scores normally within two weeks after the date of examination.

6.2 Computation of Examination Results

6.2.1 Continuous Assessment

Every course offered in the Department must have a continuous assessment component, which shall contribute 30 or 40 per cent of the total marks for the course unit at the end of the semester. Continuous assessment shall be done through essays, tests, and practical exercises. i. Scores from continuous assessment shall normally constitute 30 per cent of the full marks for courses which are primarily theoretical. ii. For courses which are partly practical and partly theoretical, scores from continuous assessment shall constitute 40% of the final marks. iii. For courses that are entirely practical, continuous assessment shall be based on a student's practical work or reports and shall constitute 100% of the final marks. In partly theoretical and partly practical courses, the 40% CA will be broken into 20% for theory and 20% for practical. The student shall be given his/her continuous assessment test scripts and marked written assignments with his/her scores on them as feedback and the same shall be published before the end-of-semester examinations. ***Any student who has no score for continuous assessment earns an F grade for the course irrespective of his/her examination score***.

6.2.2 Semester Examination

Examinations are administered at the end of each course, usually at the end of the semester. The end-of-semester examination shall contribute 60 or 70 per cent of the total marks for each course unit and credits shall be earned for course units passed. All courses shall be graded out of a maximum of 100 marks comprising: Final Examination: 60% - 70%; and Continuous

assessment (Quizzes, Homework, Tests, Practicals): 30% - 40%. Each course shall normally be completed and examined at the end of the semester in which it is offered. The Grade Point Average (GPA) will be calculated based on the total number of course units a student had registered for during the semester, whether passed or failed, and reported along with the semester and session results. However, the marks a student scores for any course unit not originally registered for shall be discarded.

6.2.3 Grade Point Average (GPA)

Grade Point Average (GPA) is a measure of the average performance of a student for a given semester expressed in grade points earned in all the course units taken by the student during the semester. The Grade Point Average (GPA) is derived from the raw scores obtained by a student in all the course units taken. It is computed by first multiplying the numerical value assigned to the letter grade attained in each course unit (that is, the *value points*) by the number of credit units assigned to that course to obtain the *grade points.* Then, these grade points are summed across the number of courses taken to obtain the *total grade points* earned in that semester. Finally, the sum - the *total grade points* earned in that semester - is divided by the total credit units the student registered for in the semester to get the grade point average. An illustration of the computation of the GPA is given below.

Table 1: An illustration of GPA Computation

Col. 1	Col. 2	Col. 3	Col. 4	Col. 5	Col. 6	Col. 7
Courses Registered	Credit Units	Raw Scores (%)	Letter Grade	Value Points	Grade Points	Remarks
SEN101	3	75	A	5	15	Grade Points Average (GPA) is derived by dividing the total of Col. 6 by the total of Col. 2, that is, GPA = Total of Grade Points (GP) earned/Total Credit units registered (TC) = 82/24 = 3.42
SEN103	2	67	B	4	8	
GST111	1	65	B	4	4	
SEN103	2	78	A	5	10	
SEN105	2	55	C	3	6	
GES 121	2	70	A	5	10	
GES 131	2	46	D	2	4	
GES 141	2	40	F	0	0	
MTH101	3	64	B	4	12	
PHY 111	2	80	A	5	10	
PHY 121	1	56	C	3	3	
TOTAL	24				82	

6.2.4 Cumulative Grade Point Average (CGPA)

The Cumulative Grade Point Average (CGPA) is a measure of a student's overall performance in his/her programme of study at the end of a particular semester or academic session. It is an up-to-date average or mean of the cumulative grade points (CGP) earned by a student at any point in his/her programme of study. The CGPA is derived by adding the *grade points* earned by the student in each course taken to-date; then, this aggregate quantity (the cumulative grade points)

is divided by the sum of the credit units of all the courses the student has taken to-date to obtain CGPA = (cumulative grade points) - (cumulative credit units).

6.3 External Examiner System

At the beginning of each semester, the external examiner shall vet the course outlines for each course offered at the 400 level for the programme for which he/she is responsible; vet the questions to be given at the end-of-semester examination as well as the marked scripts for the end-of-semester examinations. The external examiner responsible shall also participate in final year project defence and in the determination of overall results/ classification of degrees.

7. Department Boards

The Department Boards comprise Departmental Board of Studies and Departmental Board of Examiners.

1. A Departmental Board of Studies which, consists of all course lecturers and chief technologist in the Department. It shall deliberate and make recommendations to the Faculty Board of Studies on all matters relating to academic programmes of the Department.
2. A Departmental Board of Examiners consisting of all course lecturers of the Department. Departmental Board of Examiners shall deliberate and make recommendations to the Faculty Board of Examiners on all matters relating to examinations.

The Head of Department chairs the two boards.

8. Withdrawal and Probation

i. A student whose CGPA falls below 1.50 at the end of a session shall be placed on a period of probation for one academic session.
ii. At the end of a particular period of probation, if the student still earns a CGPA of less than 1.50, he/she shall be required to withdraw from the programme.
iii. In order to minimize waste of human resources, consideration is normally given to students withdrawn from 200 level of the programme to transfer to other related programmes within the Faculty of Natural Sciences or to any other programmes in other Departments within the University perceived as relevant for the purpose.

9. Repeating Failed Courses/Excess Credit Units

Subject to the conditions for withdrawal and probation, a student may be allowed to repeat the courses failed at the next available opportunity provided that the total number of credit units he/she has to carry over during that session shall not exceed 48. The Grade Points earned in all the attempts in such courses will count towards the CGPA. However, final year students can apply to Senate through their Head of Department to carry an excess of six (6) Credit Units if that will facilitate their graduation within a given academic session.

10. Class Attendance

For any student to be eligible to sit for an examination, he/she must have 80% class attendance. A list of registered students in the faculty shall be made available to students

present in the lecture/practical/seminar to sign. Lecturers are responsible for collecting the signed attendance sheets and submitting them to the Faculty Officer. Percentage attendance would be collated at the end of the semester by the Department before the examination. Heads of Department must update the attendance of students to advise them accordingly.

10.1 Absence from Lectures or Examinations

i. Permission to be away from lectures for whatever reason must be sought in writing from the Head of Department

ii. When a student misses an examination, he/she is advised to apply in writing to the Senate through the Head of Department and Dean to be granted an opportunity to take the examination. If the reason for missing such examination is cogent enough, Senate may approve the application for the student to take the examination at the next available opportunity.

iii. In case of ill-health, a student shall provide a medical report issued by an authorized medical practitioner and counter-signed by the Medical Director of the Medical Centre of Veritas University, Abuja.

10.2 Suspension of Studies

No matriculated student of the university may keep away from studies for any reason without informing the university authority through the Head of Department and the Registrar. The information **Must** be in writing and stating reasons.

11. Academic Advisers

Academic advisers are assigned to students in each academic session to counsel them on their academic performances. Students are free to meet with their academic adviser at least once in the semester for counselling especially prior to course registration. The students could also meet the HOD who is readily available and accessible to look into their problems.

12. Senate Decisions on Forgery

Forgery is viewed by Senate as a very serious breach of student matriculation oath and the rules and regulations of the University. The punishment for this offence is expulsion (dismissal) from the university. Forgery of any University document attracts same punishment.

13. Grading System

The grading system of the University is as follows:

		Grades	**Grade Points**
70% - 100%	- - -	A	5
60% - 69%	- - -	B	4
50% - 59%	- - -	C	3
45% - 49%	- - -	D	2
0% - 44%	- - -	F	0

14. Graduation Requirements

1. O' level credit requirement for entry to the programme may be deferred (to be passed before graduation or before issuance of degree certification) for the following subjects: Physics.
2. Minimum number of Earned Credit hours for graduation: 147
3. Minimum number of years for graduation (UTME Candidates): 4
4. Minimum number of years for graduation (Direct Entry Candidates): 3
5. Residency requirement (if any): minimum of 4 years – in-campus residence – and maximum of 6 years
6. Minimum CGPA for graduation: 1.50/ (Maximum: 5.00)
7. Other requirements (please specify): Nil

15. Award of Degree

The type of degree awarded to a student will be determined by the final Cumulative Grade Point Average of the student in his/her final year as shown below:

Class of Degree	CGPA Required
First Class	4.50 - 5.00
Second Class Upper Division	3.50 - 4.49
Second Class Lower Division	2.40 - 3.49
Third Class Lower	1.50 - 2.39

15.1 Title of Degree

The Software Engineering programme leads to the **BSc Software Engineering** Degree.

15.2Withdrawal of Degrees

All Degrees awarded by Veritas University; Abuja remain the property of the University. The University reserves the right to withdraw or effect correction at any time to a degree that was awarded in error.

16. Nigeria Association of Computing Students (NACOS)

The Department houses the Veritas University Chapter of the Nigeria Association of Computing Students (NACOS). A registered student of the Department is automatically a member of NACOS whose motto is: **Towards Advanced Computing**. The association annually elects its executives, headed by a president who is thus intermediary between students and departmental management. A student of the Department is strongly enjoined to partake in the association's activities.

NACOS shall be organised into clubs such as Coding club, Tutorials club, Start-up club, etc, with Coordinators appointed for each.

There shall be an Annual Prize for Innovation (API) organised by NACOS every academic session. The prize will be sponsored by NACOS in collaboration with consulted organizations and companies. The event shall be well publicized with a lot of invitations made by NACOS from both within and outside the university. All NACOS members from 100 to 400 Levels will

be encouraged to participate in the annual exhibition and competition. A panel of seasoned evaluators will assess the exhibited and presented apps based on certain criteria. These criteria include among others: creativity, innovation, originality, novelty, professionalism, market value, practical relevance, etc.

NACOS shall encourage its members to engage in highly prized software projects, and also initiate and incubate sustainable software start-ups.

In addition, NACOS shall encourage excellence in academics among her members through involvements in tutorial activities especially in challenging courses. Tutorial coordinators per course shall be appointed to facilitate this.

NACOS shall also organize social and sporting events as well as excursions to educative sites of interests for the benefit of its members from time to time.

NACOS shall organize an annual industrial Expo event for captains of software industries to educate students of industrial expectations, and experiences. The forum will be used to solicit for placements for industrial attachments (internships) for students during their SIWES and possible employment after graduation.

NACOS shall organize workshops, seminars and conference from time to time to educate its members and to get them well informed and abreast of current happenings in computing and industry.

NACOS shall be involved in consultancy and marketing drives to market their software products and to provide computing solutions to interested individuals, companies, organizations, and customers beginning with the university. The university shall be one of NACOS biggest clients. To this end, marketing coordinator(s) shall be appointed to facilitate this.

An annual association due to be determined be NACOS shall be paid by all its members to facilitate the programs and activities of the association. NACOS can also seek for external sponsorship of its programs and activities too.

NACOS shall float an annually published student peer-reviewed journal in software engineering to facilitate academic writing and research among its members. A director for studies and research shall be appointed to facilitate this. This director together with the NACOS research committee shall among other things also organize an annual student conference on software engineering. Best articles from the conference will be published in the student journal. The journal shall have a website and shall seek to be indexed in Google Scholar, etc.

NACOS shall organize training programs to educate members and to provide them with state-of-art skills in software computing innovations and software business. Resource persons can be sourced from within and outside the university and from both academia and industry. A director for training shall be appointed to facilitate this.

NACOS shall sign up and be involved in local, national and international software competitions.

NACOS shall organize an annual dinner and award night for its members.

NACOS executives shall each submit a report of their accomplishments at the end of their tenure to the Department through the Staff Adviser. They shall each be judged and appraised by how much they were able to accomplish the stated NACOS objectives, programs and activities. NACOS Directors shall include among others:
1. Director of Socials and Events
2. Director of Studies and Research
3. Director of Training and Tutorials
4. Director of Consultancy and Marketing
5. Director of Innovations and Start-ups
6. Etc.

16.1 Dress Code
Students shall be on full corporate wears from Mondays to Wednesdays and on full traditional (native) wears from Thursdays to Fridays. No casual wears will be allowed. The use of wears with the departmental colours is strongly encouraged. NACOS shall from time to time suggest for production branded outfits with the departmental or NACOS colours for her members.

16.2 Departmental Colours
The departmental colours include: Oxblood, Army green and White as shown in the departmental logo. The White colour symbolizes truth, simplicity, serenity, &faithfulness; the Army Green colour symbolizes productivity, innovation, creativity, &professionalism; and; the Oxblood colour symbolizes sacrifice, hard work, commitment, diligence, & doggedness. All these qualities reflect the very spirit of the Department of Software Engineering (The Centre of Excellence).

16.3 DSE Coding Club & Coding Academy
The department hosts a coding club for interested students as an extracurricular activity. The club also doubles as a coding academy. The goal of the club is to stimulate coding activity among students geared towards intensively getting them ready for industry and professional career in software engineering. The club is project-driven as it prepares students for software competitions and certifications through rigorous peer training. The club is entirely student-driven but however supervised by a staff adviser who serves as a coordinator and coach. Membership is voluntary and open to all students in the department.

16.4 Software Exhibition and Competition
The department holds an annual software exhibition and competition. This event takes place every second semester of every session. The event enables students to showcase all the software products they have developed in the last one year. Prior to the event and over the year, students are mentored, trained and prepared by staff members of the department in preparation for the event. The Vice Chancellor’s Annual Prize for Innovation (API) will be contested for and won at the event.

17. Regulation on the Use of Laboratory

ALL students shall have good personal laptops for private practice and personal practical sessions.

Students coming into the Software Engineering Laboratory for Lab work are expected to observe in good faith, the following laboratory regulations for proper usage and management of the laboratory:

i. Eating or drinking of any kind and at any time while in the laboratory is prohibited
ii. Students are not expected to litter the laboratory space.
iii. Playing or fighting inside the laboratory is prohibited
iv. Buying and selling inside the laboratory is not allowed at any time
v. No student is allowed to remove any equipment from the laboratory
vi. No student is allowed to remove laboratory stool/seat from the laboratory
vii. Damage of any laboratory equipment must be promptly reported.
viii. Stealing of any lab item is a punishable offence.

18. SIWES

The SIWES exercise holds in the long vacations at the end of 200 and 300 levels. Each student while on SIWES must target at carrying out and completing at least one full software project for presentation as part of his or her SIWES project. This is in addition to any other work experience report to be presented. Students should let their supervisor(s) at their places of placement know that the software project is one of the goals of our SIWES programme. Where the place of internship is not supportive, the students should still carry out the project all the same. The SIWES project name should be transmitted to the SIWES coordinator on commencement of the SIWES programme.

19. Final Year Students' Project

All students' final year software applications must be professionally engineered from requirements to design to coding to testing and possibly deployment and usability testing using suitable software engineering methodology before project defence. The project should be research-based, novel, and challenging and should be a solution to a practical problem.

20. Course Structure, Content and Synopsis

100 LEVEL: FIRST SEMESTER

Course	Status	Course Title	Contact			Unit(s)
			L	T	P	
SEN101	C	Introduction to Computing and Applications	2	1	3	3
SEN103	C	Interaction Design and Usability Engineering	2	1	3	3
SEN105	C	Introduction to Software Engineering	2	1	3	3
SEN181	C	Software Engineering Lab I	2	2	3	2
GST111	R	Communication in English I	1	1	0	2
GST113	R	Nigerian Peoples and Culture	1	1	0	2
GST115	R	History and Philosophy of Science	1	0	0	2

GST121	R	Use of Library, study skills and ICT	1	0	0	2
GST 171	R	Ethics	1	0	0	0
MTH101	C	Elementary Mathematics I	2	1	0	2
PHY101	C	General Physics I	2	1	0	2
PHY107	C	General Practical Physics I	0	0	3	1
Total						24

100 LEVEL: SECOND SEMESTER

Course	Status	Course Title	Contact			Unit(s)
			L	T	P	
SEN102	C	Principles of Programming I	2	1	3	2
SEN104	C	Introduction to Web Technologies	1	1	3	2
SEN106	C	User Experience Design and Evaluation	2	1	3	2
SEN108	C	Logic and Its Application in Computer Science	1	1	3	2
SEN182	C	Software Engineering Lab II	2	2	3	2
SEN190	C	Practicum			18	1
GST112	R	Communication in English Il	1	1	0	2
GSTI22	R	Logic, Philosophy and Human Existence	1	1	0	2
MTH102	C	Introductory Mathematics Il	2	1	0	3
PHY102	C	General Physics Il	2	1	0	2
PHY108	C	General Practicals Physics Il	0	0	3	1
GST 124	R	Communication in French	1	1	0	2
GST142	R	Community Service	0	0	1	1
Total						24

200 LEVEL: FIRST SEMESTER

Course	Status	Course Title	Contact			Unit(s)
			L	T	P	
SEN201	C	Discrete Structures	2	1	0	3
SEN 203	C	Software Requirements and Design	2	1	3	2
SEN205	C	Computer Architecture and Organisation	2	1	3	3
SEN207	C	Data Structures and Algorithms	2	1	3	3
SEN209	C	Information Architecture	2	1	3	2
SEN281	C	Software Engineering Lab III	2	2	3	2
GST211	R	Basic Spiritual Theology	1	0	0	0
GST223	R	Introduction to Entrepreneurship	2	0	0	2
MTH203	R	Linear Algebra I	1	1	0	2
Total						19

200 LEVEL: SECOND SEMESTER

Course	Status	Course Title	Contact			Unit(s)
			L	T	P	
SEN202	C	Principles of programming II	2	1	3	3
SEN204	C	Software Construction	1	1	3	2
SEN206	C	Design and Analysis of Computer Algorithms	1	1	3	3
SEN208	C	Principles of Operating Systems	1	1	3	2
SEN210	C	Software Engineering Process	2	1	3	2
SEN212	C	Information Visualization	2	1	3	2
SEN282	C	Software Engineering Lab IV	2	2	3	2
SEN290	C	SIWES I			18	3
GST221	R	Peace Studies & Conflict Resolution	2	0	0	2
GST272	R	Social Teachings of the Church	1	0	0	0
Total						19

300 LEVEL: FIRST SEMESTER

Course	Status	Course Title	Contact			Unit(s)
			L	T	P	
SEN301	C	Objected-oriented Analysis and Design	2	1	3	3
SEN303	C	Software Testing and Quality Assurance	1	1	3	2
SEN305	C	Web Application Development	2	1	3	3
SEN307	C	Database Systems	2	1	3	3
SEN309	C	Concepts of programming Languages	1	1	3	2
SEN381	C	Software Engineering Lab V	2	2	3	2
GST311	R	Intro. to Entrepreneurship Studies	0	0	9	2
STA343	C	Operation Research I	2	1	0	3
Total						21

300 LEVEL: SECOND SEMESTER

Course	Status	Course Title	Contact			Unit(s)
			L	T	P	
SEN302	C	Software configuration management and maintenance	1	1	3	2
SEN304	C	Software engineering project management	1	1	3	2
SEN306	C	Research Methodology	1	1	0	1
SEN308	C	Software Engineering Professional Practice	1	1	3	2
SEN310	C	Software Engineering Security	1	1	3	2
SEN312	E	Al and Expert Systems	2	1	3	3
SEN314	E	Engineering Mobile Applications	1	1	3	2
SEN316	E	Embedded Systems	2	1	3	2

SEN382	C	Software Engineering Lab VI	2	2	3	2
SEN390	C	SIWES II			18	3
Total						21

400 LEVEL: FIRST SEMESTER

Course	Status	Course Title	Contact			Unit(s)
			L	T	P	
SEN401	C	Software engineering economics	1	1	3	2
SEN403	C	Human-computer interaction	1	1	3	2
SEN405	C	Open-Source Software Development and Applications	1	1	3	2
SEN407	C	Distributed, Parallel, and Cloud Computing	1	1	3	2
SEN409	C	Software Architecture and Design	2	1	3	2
SEN411	E	Special Topics in Software Engineering	2	1	3	2
SEN413	C	Research Seminar	1	0	1	1
SEN415	C	Visual Design	2	1	3	2
SEN481	C	Software Engineering Lab VII	2	2	3	2
Total						17

400 LEVEL: SECOND SEMESTER

Course	Status	Course Title	Contact			Unit(s)
			L	T	P	
SEN402	C	Enterprise Application Development	2	1	3	2
SEN404	C	Virtual and Augmented Reality	2	1	3	2
SEN406	E	Fault—Tolerant Computing	2	1	0	2
SEN408	E	Game Design and Development	2	1	3	2
SEN410	E	Modelling and Computer simulation	1	1	3	2
SEN482	C	Software Engineering Lab VIII	2	2	3	2
SEN490	C	Project	0	0	18	6
Total						18

L = Lecture Contact Hours; T= Tutorials; P Practical Contact Hours; C — Compulsory; R = Required; E = Electives.
*XYZ represents the three-letter code for each programme.
** The status of each of these courses depends on the programme under consideration

NOTE
C = Compulsory
E = Elective
R = Required
L = Lecture Hours
P = Practical Hours
T = Tutorial Hours

MIXED MODE COURSES (THEORY AND PRACTICAL): TUTORIAL, LECTURE AND PRACTICAL COMPONENTS:

The Tutorial components will be handled by Graduate Assistants (GAs) or Tutorial Assistants (TAs) under the supervision of the lecturer for the lecture component while the practical components will be handled by the course lecturers in charge of the practical components in collaboration with the Laboratory Technologists. In the absence of GAs or TAs, the tutorial component will be assigned to a lecturer who will play a supportive role to the lecture component of the course, otherwise, the tutorial component will be handled by the lecturer handling the lecture component for the course. The overall continuous assessment for the course will include lecture, tutorial and practical aspects of the course. Also, the final examination assessments will include theory and practical based assessments. The theory component of the course will constitute 50% of the overall examination score while the practical component will constitute 50% of the overall examination score. All theory or practical examinations will last for between 3-4 hours and will have 5 or 6 questions out of which students are to select 3 or 4 to answer depending on the credit load of the course. To be precise, question papers for two (2) unit courses will include 5 questions out of which students will pick 3 to answer and the time allowed for this set of question papers is 2 hours. Also, for three (3) unit courses, the question papers will include 6 questions out of which students will answer 4 and the time allowed for this set of question papers is 3 hours. For practical exams (hands-on): question papers for two (2) unit courses will include 5 questions out of which students will attempt 3 and the time allowed for this set of question papers is 3 hours. In addition, for three (3) unit courses, the question papers will include 6 questions out of which students will answer 4 and the time allowed for this set of question papers is 4 hours. All practical examinations will be hands-on and will take place in the lab. Examination questions (theory or practical) will cover all aspects of the course. The continuous assessments scores for both theory and practical components will be 20 marks each while the examinations will be 30 marks each. Tutorial component will be assessed with a continuous assessment score of 10 marks, leaving the CAs for the Lecture components to be 10 marks, (e.g., **Theory:** Tutorial CA: 10 marks, Lecture CA: 10 marks, Written Exams: 30 marks; **Practical:** CA: 20 marks, Project (as part of Exams): 10 marks, Practical Exams: 20 marks.)

In addition, for the practical components, each student will undertake a practical term project either individually or as a group, under the mentorship of the course lecturer (practical component). This project will be submitted and defended at the end of the semester. However, the students' progress in the projects will be monitored throughout the semester. The project must be related to the course. The project will be scored 10 marks and will contribute to the practical examination overall score. This project is not part of the continuous assessment but part of the examination. Students should not use the same project in more than one course.

Lab Manuals: The Lab manuals and lab sessions are structured to consist of 10 lab sessions; however, the lab sessions do not terminate at the end of the 10^{th} lab session. There are 14 weeks of academic activities and so, the 11^{th} to the 14^{th} lab sessions will be used for students' projects, supervision, presentations, and defence for the various practical courses. So, the lab does not come to an end after the 10^{th} lab session, but continues with intensive students' projects, supervision, presentation, and defence. The provision of 10 lab sessions in the lab manual is to give space and allowance for students to carry out their projects, and for lecturers to closely

supervise the projects, and also for the students to do their presentations and practical defence. So, the lab sessions continue to the 14th week of the semester.

SOFTWARE ENGINEERING LABS COURSES:

These are the full practical course components of the software engineering programme (practical only courses). They have two hours each for lectures and tutorials each week for the teaching of the theoretical aspects of the lab Practical and projects. Then the full hands-on practical sessions have three (3) hours weekly. Adequate time and attention shall be given the Lab sessions. Lecturers are to spend the full three (3) hours Lab time period with the students in the Lab. There shall be no written examinations for these courses, however students shall be assessed based on their projects' outputs. The lab continuous assessments (CAs) will cover students' active and continuous lab participations, attendance, and involvement in lab exercises (30%) while 50% will be for the finished term project which will be evaluated by a panel of assessors for creativity, novelty, professionalism, practicality, market value, innovation, originality, and relevance, etc. The Tutorial and Lecture Components of the SE Lab will be assessed 20% (In summary, Tutorial (10 marks), Lecture (10 marks), SE Lab CAs (30 marks), and SE Lab Project (50 marks)).

FULL THEORY COURSES:

These group of courses will be taught as both tutorial and lecture components (theory only courses). The assessments will also be 100% theory. The assessment scores are broken down as follows: Tutorial component CAs (10 marks), Lecture component CAs (20 marks), and final examinations (70 marks). Question papers for the final examinations for courses with two (2) units will include 5 questions out of which students will pick 3 to answer and the time allowed for this set of question papers is 2 hours. However, for three (3) unit courses, the question papers will include 6 questions out of which students will answer 4 and the time allowed for this set of question papers is 3 hours.

LAB DECORUM AND ETIQUETTE

The lab environment must be purely dedicated to lab practical and other lab related activities. Every lab session must be serene and devoid of distractions, noise, rowdiness, and frivolities. Students who are not part of a given lab session must be excused out of the lab while the given lab session lasts. The lab is only for students participating in a given lab session. All students for a specified lab must be adequately prepared for the lab session before the session begins. They are to come in or bring with them all materials, tools, and equipment needed for the given lab session. They are to be properly dressed following the departmental dress code for the day. All lab activities and behaviours inimical to the lab such as playing with phones, careless browsing, making noise, distracting others, etc., must be avoided. Lecturers on the other hand are to be adequately prepared for every practical and lab session and are to make every practical and lab session hands-on, demonstrative, interactive, attractive, appealing, participatory, and engaging for the students. Each lab or practical session lasts for three (3) hours. Students are to be meaningfully engaged with lab activities while the lab/practical session lasts. All practical and lab courses are project based, project oriented and project driven.

DUTIES OF THE SOFTWARE ENGINEERING LABORATORY TECHNOLOGIST

S/N	Duties and Responsibilities
1.	Manage the SE Labs following SE standards
2.	Prepare/setup the lab and make it ready for each lab session
3.	Guide students in their lab practical and assessments
4.	Collect students' lab manuals weekly for assessment by the course lecturer
5.	Ensure that each lab session maintains its decorum and serenity.
6.	Be knowledgeable in each lab activity so as to effectively and efficiently be of help to students during each lab session
7.	Be exemplary, motivational, professional, faithful, committed and charismatic
8.	Be able to carry out a lab session in the case of the absence of a course lecturer
9.	Suggest laboratory needs and upgrades to the department
10.	Assist in the preparation and upgrade of lab manuals
11.	Assist in the acquisition of lab equipment, materials and accessories (software & hardware).
12.	Assist the course lecturers in all SE laboratory related functions
13.	Be punctual and regular in the lab
14.	Maintain the lab and regularly report to the department on the state of the lab
15.	Ensure the lab is state-of-the-art following cutting edge academic and industry standards
16.	Maintain an enduring lab culture for the department
17.	Be creative and suggest innovations in the lab
18.	Lead students in developing innovative applications
19.	Help in preparing students for software competitions
20.	Be capable of guiding and supervising students on internships in the lab
21.	Be involved in SE research in the lab. Be available and assist in SE studies being carried out in the lab
22.	Follow SE best practices in the handling of the SE labs

COURSE SYNOPSES

100LEVEL FIRST SEMESTER

SEN101: Introduction to Computing &Applications (3 units)

Lecture and Tutorial Components: Introduce: l. The computer as a tool for content creation, storage, and processing, 2. Some of the broad fields of computer sciences from programming languages to Artificial Intelligence, 3. Computing methods to problem-solving and program. Emphasis is placed on gaining literacy and some practice with computers, applications and problem-solving, and the broad areas of computing.

Practical Component (Use of programming languages to solve algebraic operations, mathematical, statistical, common, and everyday problems, recursive programming). Suggested language: Python, Java, C, C#, C++.

SEN103: Interaction Design and Usability Engineering (3 unit)

Lecture and Tutorial Components: Meaning of human-computer interaction, applied ergonomics; user-centered design, interaction design, engineering usability: models, methods

and metrics; usability dimensions, software technology adoption. Lab and field-based Usability Testing and evaluation.

Practical Component (Designing interactive interfaces with Adobe XD, Sketch, and Figma using interaction design principles. Practical frontend designs and usability evaluations; Development of simple applications with good interactive interfaces and usability; development of simple applications using user-centered design and interaction design principles). Suggested Tools: Adobe XD, Figma, Sketch, ConceptDrawMindMap, Adobe Illustrator, Enterprise Architect, HTML, CSS, JavaScript, PHP, Java Web, and Python

SEN105: Introduction to Software Engineering (3 units)

Lecture and Tutorial Components: Software engineering concepts and principles; design, development, and testing of software systems. Introduction to the software life cycle. Requirements, design, and testing. Review of principles of object orientation. Object-oriented analysis using UML. Frameworks and APIs. Introduction to a client-server architecture. Analysis, design, and programming of team projects including user interface considerations.

Practical Component (Planning, Requirements, Design, Implementation, Testing, and Deployment: Engineering a simple application from start to finish using state-of-the-art software engineering principles). Suggested Tools: PHP, Python, C, C#, C++, Java, StarUML, E-Draw, Microsoft Visio, ConceptDrawMindMap, Adobe Illustrator, Enterprise Architect, Selenium, Telerik Test Studio.

SEN181: Software Engineering Lab I (2 units)
(General Programming Techniques Projects)

Simple algorithms – Fibonacci, Tower of Hanoi, Game of Life; simple applications – calculator, games (chess), payroll and grading systems, To-do-list, Notepad, etc.

GST111: Communication in English I: (2 units)

Effective communication and writing in English Language skills, essay writing skills (organization and logical presentation of ideas, grammar, and style), comprehension, sentence construction, outlines, and paragraphs.

GST113: Nigerian Peoples and Culture: (2 units)

Study of Nigerian history, Culture, and arts in pre-colonial times, Nigerian's perception or his world; Culture areas or Nigeria and their characteristics: Evolution of Nigeria as a political unit: Indigene settler phenomenon: Concepts of trade; Economic self-reliance, social justice; Individual and national development, norms and values; Negative attitudes and conducts (cultism and related vices); Re-orientation of moral; environmental problems.

GST115: History and Philosophy of Science: (2 units)

Scientific evolution of man: the history of science; classification; scientific methods. Science and the environment: terrestrial and cosmic life; ecology and types of habitats; climate and vegetation; Energy resources: fossil fuels, nuclear energy and renewable energy resources; the

Nigerian energy reserves. Characteristics of living things; cell and tissue biology, biochemistry and cellular metabolism, taxonomy of living things; heredity and evolution: Technology and technological evolution; Engineering technology and socio-economic development of Nigeria; the effect of adequate and stable electric power supply and satellite communication technology on the technological development of Nigeria; political and other constraints. The interaction between science and technology: Social implications of advances in science and technology: automated industrial plants; satellite technology; space technology; genetic engineering technology; weapons of mass destruction (WMD), and environmental pollution.

GST121: Use of Library, study Skills and ICT: (2 units)

Brief history of libraries; Library and education; University libraries and •other types of libraries; Study skills (reference services); Types of library materials, using library resources including e-learning, e-materials, etc; Understanding library catalogues (card, OPAC, etc) and classification; Copyright and its implications; Database resources; Bibliographic citations and referencing. Development of modern ICT; Hardware technology; Software technology; Input devices; Storage devices; Output devices; Communication and Internet services; Word processing skills (typing, etc).

GES171: Ethics (0 unit)

Meaning, nature and scope of ethics; Ethics as character formation; Definition & identifying ethical values; Respect for human life and dignity of the human person; Integrity and discipline; Hard work; Materialism and the Nigerian society; Ethical issues in tertiary institutions in Nigeria; Drug abuse & cultism; Pornography & cybercrime (Social ills of info tech (IT)); Violence and Nigerian society; Social interaction & religious bigotry in the society; Sensitivity to ethnic and racial issues.General Introduction; Free will, human responsibility and moral choice; The nature of human Sexuality; Child abuse; Rape & sexual assault; The problem of teenage pregnancy & abortion; Transgender, gay, lesbianism & LGBT issues; Euthanasia and suicide; The Evil of human trafficking & modern slavery; Religious bigotry and ethnicity; The bane of corruption in Nigerian society; The value system in Nigerian Society: A review.

MHT101: Elementary Mathematics I (2 units)
(Algebra and Trigonometry)

Elementary set theory, subsets, union, intersection, complements, Venn diagram; Real numbers, integers, rational and irrational numbers. Mathematical induction, real sequences and series, theory of Quadratic equations, Binomial theorem, complex numbers, algebra of complex numbers, the Argand diagram. De-Moiré's theorem, nth roots of unity. Circular measure, trigonometric functions of angles of any magnitude, addition and factor formulae.

PHY101: General Physics I (2 units)
(Mechanics, Thermal Physics and Waves)

Space and Time, Units and Dimension, Kinematics; Fundamental Laws of Mechanics, statics and dynamics; work and energy; Conservation laws. Moments and energy of rotation; simple harmonic motion; motion of simple systems; Elasticity; Hooke's law, Young's shear and bulk

moduli, Hydrostatics; Pressure; buoyance, Archimedes' Principles; Surface tension: adhesion, cohesion, capillarity, drops and bubbles; Temperature; heat; gas laws: laws of thermodynamics; kinetic theory of gases; Sound. Types and properties of waves as applied to sound and light energies. Superposition of waves. Propagation of sound in gases, solids and liquids and their properties. The unified spectra analysis of waves. Applications.

PHY107: General Practical Physics 1 (1 unit)

This introductory course emphasizes quantitative measurements, the treatment of measurement errors, and graphical analysis. A variety of experimental techniques will be employed. The experiments include studies of meters, the oscilloscope, mechanical systems, electrical and mechanical resonant systems, light, heat, viscosity, etc., covered in PHY101 and PHY102. However, emphasis should be placed on the basic physical techniques for observation, measurements, data collection, analysis and deduction.

100 LEVEL SECOND SEMESTER

SEN102: Principles of Programming 1 (2 units)

Lecture and Tutorial Components: Introduces the basic principles of programming and the fundamentals of procedural, GUIs and object-oriented program using a suitable high-level object-oriented language. Emphasis should be placed on such principles as procedural and data abstraction, encapsulation, code reuse and composition. Amble programming labs and projects form part of this course.

Practical Component (Use of programming languages to model problems as objects and solving them in an Object-Oriented manner.) Suggested language: Java, Python, C#, C++.

SEN104: Introduction to Web Technologies (2 units)

Lecture and Tutorial Components: Broadly examines the key technologies and programming models of the web and its underlying Internet infrastructure including client-side technologies such as HTML, CSS and JavaScript, content format and server-side technologies such as web and application servers, database back ends, client-server programming model, communication protocols such as http and TCP/IP. Students should also be introduced to content creation with client-side technologies and time permitting, some basic database driven application using a familiar application framework.

Practical Component (Use of HTML, CSS, Bootstrap, MySQL database, C#, C++, ASP.NET, JavaScript (React.js, Node.js), PHP (Laravel), Java Web, and Python (Flask/Django) for building novel applications)

SEN106: User Experience Design and Evaluation (2 units)

Lecture and Tutorial Components: Concept of experience, user experience (UX), aspects of user experience, instrumental & non-instrumental UX qualities, interaction design, user-centred design, design heuristics, UX design principles and guidelines; pragmatic, aesthetic, affective, sensitive, social, inclusive and hedonic designs. UX design and evaluation for wearable, mobile, and desktop applications. User experience frameworks and models.

Practical Component (Low-fidelity and high-fidelity prototyping using Adobe XD, Sketch and Figma. Practical user interface (UI) designs and user experience (UX) evaluations; Development of simple applications with interfaces that offer impressive user experiences; development of simple applications using user experience design principles). Suggested Tools: Adobe XD, Sketch, Figma, Adobe Illustrator, Enterprise Architect, HTML, CSS, JavaScript: Required Materials: drawing book, sticky note pack (with multiple colours), cardboard sheets.

SEN108: Logic and its Applications in Computer Science (3 units)

Lecture and Tutorial Components: Topics include: valid & invalid arguments; translating from English to the language of propositional and predicate logic; formal deduction and its role in proving the validity of an argument; logic & computer science (computing)— digital logic; how to build circuits from logic gates and how to minimize circuits using propositional logic; introduction to Prolog (or any related language) - a Programming language based on logic; and, the applications of logic in computer science - Al, automated theorem-provers, expert systems, and so on. Alternatively, Python can also be used.

Practical Component (Digital logic designs and simulations; design, development and deployment of simple intelligent and expert systems). Suggested Tools: Python, Java, C, C#, C++, Digital trainer, Xilinx.

SEN182: Software Engineering Lab II (2 units)
(Web Apps and Mobile Apps Projects)

Electives 1: Web Apps: Building simple web pages with HTML, Styling pages with CSS, exploring HTML and CSS frameworks, Using JavaScript for web apps, exploring backend functionality and frameworks – sessions, building databases, authentication and authorization, CRUD (Create, Read, Update and Delete) operations, Web services and APIs (Application Programming Interface). Suggested Tools: Java Web, Perl, JavaScript (React.js/Node.js), PHP (Laravel), Python (Flask/Django), C#.

Elective 2: Mobile Apps: Running database apps on mobile, creating games for mobile, User Interface, Interaction and User Experience on Mobile, Working with Graphics and Camera, Development of simple mobile apps – native & hybrid (Voice recognition, Chat Bots, Utility apps). Utilizing mobile sensors for developing apps. Suggested Tool: React native, Dart, Flutter.

SEN190: Practicum (1 unit)

This is a first-level practical internship for SE students. Students are expected to do an industrial attachment with a software industry to understand and familiarize themselves with the software engineering expectations and demands in the industry. They are also expected to produce a project (on software development) that demonstrates practically a first-year software engineering experience. The software developed must be original and novel. Students should get their projects approved by the practicum coordinator before commencing the projects. Students will write a practicum report and also present and defend their practicum experience and projects. This practicum will take place during the long vacation holiday and will last for three (3) months (12 weeks). The assessment for this course is as follows: Software Project (50

marks), Logbook (20 marks), Practicum Experience Report (15 marks), Industry Supervisor's Report (15 marks). However, since this course is a year one industrial exposure and training course, the industry scenario can be mimicked and modelled (simulated) in the department by assigning students to staff members with industry experience to supervise students in their year one industrial training and practicum.

GS'I'122: Logic, Philosophy and Human Existence: (2 Units)

A brief survey of the main branches of Philosophy; Symbolic logic; Special symbols in symbolic logic-conjunction, negation, affirmation disjunction, equivalent and conditional statements, and the law of tort. The method of deduction using rules of inference and bi-conditionals, qualification theory. Types of discourse, nature or arguments, validity and soundness, techniques for evaluating arguments, the distinction between inductive and deductive inferences; etc. (Illustrations will be taken from familiar texts, including literature materials, novels, law reports, and newspaper publications).

GST112: Communication in English II: (2 Units)

Logical presentation of papers; Phonetics; Instruction on lexis; Art of public speaking and oral communication; Figures of speech; Précis; Report writing.

GST124: Communication in French (2 units)

Introduction to French, Alphabet, and Numeracy for effective communication (written and oral). Conjugation and simple sentence construction based on communication approach, sentence construction, comprehension, and reading of simple texts.

GES142: Community Service (1 units)

Civil works beneficial to the University community and its environs including but not limited to farming, road building and maintenance, landscaping, planting of flowers and hedges, grass-cutting and general cleaning of campus and its environs, concreting and laying of seating and footpath slabs, etc.

MTH102: Elementary Mathematics II (3 units)
(Calculus)

Functions of a real variable, graphs, limits, and idea of continuity. The derivative, as limits of the rate of change. Techniques of differentiation, maxima, and minima. Extreme curve sketching, integration, Definite integrals, reduction formulae, application to areas, and volumes (including approximate integration: Trapezium and Simpson's rule).

PHY102: General Physics Il (2 units)
(Electricity, Magnetism and Modern Physics)

Electrostatics: conductors and currents; dielectrics; magnetic fields and electro-magnetic induction; Maxwell's equations; electromagnetic oscillations and waves; Coulomb's law;

methods of charging; Ohms law and analysis of DC circuits; AC voltages applied to induction, capacities, and resistance; applications.

PHY108: General Practical Physics Il (1 unit)

This is a continuation of the experiments designed for PHY101 and PHY102 some of which have been covered under PHY107.

200 LEVEL FIRST SEMESTER

SEN201: Discrete Structures (3 units)

This covers the mathematics needed for computer science and software engineering. Topics covered include functions, relations, propositional and first-order predicate logic, set theory, proofs and their construction, counting, and elementary probability. Application of discrete structures to computing.

SEN203: Software Requirements and Design (2 units)

Lecture and Tutorial Components: Definition of a software requirement, product and process requirements, functional and non-functional requirements, emergent properties, quantifiable requirements, and system and software requirements; Requirement's process — process models and actors, process support and management, and process quality and improvement; Requirements elicitation-requirements sources and elicitation techniques.Requirements elicitation, analysis, specification, documentation, verification and validation, and design modeling.

Practical Component (Textual requirements (functional and non-functional requirements), design with UML modeling (analysis phase): user case diagram, activity diagram, sequence diagram, collaboration diagram, state machine diagram, class diagram, entity relationship diagram, data flow diagram). Suggested Tools: StarUML, Edraws, ConceptDrawMindMap, Adobe Illustrator, Enterprise Architect, Jira, Confluence, etc.

SEN205: Computer Architecture & Organization (3 units)

Lecture and Tutorial Components: Differences between computer architecture and computer organization, the basic structure of computers: performance evaluation: metrics and calculations, performance equations, Amdahl's: CPU organization and micro-architectural level design; Instruction set design: register transfer; RISC design principles; data-path design; controller design; memory system; addressing; micro programming; computer arithmetics; survey of real computers and microprocessors; peripheral devices and input/output busses; and introduction to parallel computing. The course is a broad introduction to all aspects of computer systems organization and architecture and serves as the foundation for subsequent computer systems courses.

Practical Component (Primary components: for PCs, laptops, tablets and smartphones, etc: CPU, RAM/ROM, Firmware, motherboards, storage technologies, peripherals, microcontrollers,

networking – LAN, WAN, intranet, Internet; networking systems and workstations) Suggested Tools: Soldering iron, crimping tool, glue gum, heat sink paste, hot air blower, voltmeter, etc.

SEN207: Data Structures and Algorithms (3 units)

Lecture and Tutorial Components: Covers Abstract Data Types (ADTs) and their support and implementations in object-oriented languages. Topics include recursion, complexity analysis, linear data structures (stacks, queues, priority queues, lists, and strings), non-linear data structures (hash tables, binary trees, search trees, balanced trees, heaps), searching and sorting algorithms, and graph algorithms. This will also include substantial programming assignments and projects. Introduction to the algorithm for parallel & distributed computing.

Practical Component (Implementing and engineering data structures and algorithms using a suitable language; Developing applications built with data structures). Suggested Language: C++, Java, Python, Node.js, PHP

SEN209: Information Architecture (2 units)

Lecture and Tutorial Components: History and definition of Information Architecture (IA); Types of Information Architecture; User-Centered Design; Information Architecture Models and case studies; Information Architecture Problem Solving; Information Design; Information Architecture Development Process; Professional Practice (working as an information architect).

Practical Component (Design and organization of interfaces: Prototypes and Implementations. Information architecture for desktop, smartphone, and tablet interfaces on web and mobile platforms using Adobe XD, Sketch, and Figma. Practical information architecture for websites and other interfaces for real applications; Development of simple applications with good information architecture). Suggested Tools: Adobe XD, Sketch, Figma, EDraw, Adobe Illustrator, Enterprise Architect, HTML, CSS, JavaScript, PHP, C#, C++ and Python: Required Materials: drawing book, sticky note pack (with multiple colors), cardboard sheets.

SEN281: Software Engineering Lab III (2 units)
(Artificial Intelligence, Machine Learning, Deep Learning, Internet of Things Applications, Distributed Systems Projects, and Microcontroller Enabled Apps Projects)

Elective 1 (AI, ML & DL Projects): Machine learning: Supervised learning, Unsupervised learning and Deep learning. Artificial intelligence: the application of machine learning and deep learning in the development of intelligent machines and smart applications. Contrastive learning. AL, ML & DL algorithms; Text processing, image processing, voice recognition, pattern recognition, computer vision, data science, data mining, big data, smart environments applications, etc. Recommender systems, Chatbots, Store sales Forecasting, Time series, Retrieval based systems, Clustering systems, Weather Prediction systems, automated grading systems, natural language processing, and image processing applications, anthropomorphic smartphone assistants, etc., Implementing AI on web, mobile, networks, games, wearable devices and machine systems, etc. The goal of the projects in this course is the development of

AI applications and machines using machine learning and deep learning principles. All students' projects should be AI-based.

Elective 2 (IoT Projects): Construction and development of IoT systems and applications. Developing internet of everything related applications—smart homes, smart cities, smart traffic light systems, smart farms, smart campuses, smart offices, smart manufacturing industries, smart automobiles and machines, smart hospitals, smart banks, smart businesses, smart shops and supermarkets, etc. Suggested Tools: Arduino MCU, TinkerCAD, etc.

Elective 3 (Distributed Systems Projects): Development of distributed systems applications. Web services – RESTful and SOAP APIs; novel social media apps, telecommunication apps, video conferencing apps, cloud apps, mobile apps, online games, collaborative apps, network apps, and other distributed applications.

Elective 4 (Microcontroller Enabled Apps Projects): Microcontroller programming, programming hardwires and sensors, development of embedded and robotic systems, and other hardware-related systems. Suggested tools: MATLAB, Arduino MCU, C, C++, micro python, Simulink, Proteus & Wokwi Simulators, TinkerCAD, etc.

MTH 203 Linear Algebra I: (2 units)

Pre-requisite -MTH101,102
Co-requisite -MTH203

Vector space over the real field. Subspaces, linear independence, basis and dimension. Linear transformations and their representation by matrices - range, null space, rank. Singular and non-singular transformation and matrices. Algebra of matrices.

GES211: Basic Spiritual Theology (0 unit)

Meaning, nature, and scope of spirituality; Christian virtues; Cardinal virtues; Theological virtues; Prayer in Christian spirituality; Forms and effects of prayer; The Lord's prayer; Sin and its consequences; Conversion and spiritual growth; Christian conscience; Devotion to Mary; Charismatic movement; Spiritual direction and spiritual exercise.

GES221: Peace Studies and Conflict Resolution (2 units)

Basic concepts in peace studies and conflict resolution. Peace as a vehicle of unity and development, conflict issues, types of conflicts e.g., ethnic/religious/political/economic conflicts. Root causes of conflict and violence in Africa, elements of peace studies and conflict resolution developing a culture of peace, peace meditation, and peacekeeping, alternative dispute resolution, dialogue/Arbitration in conflict resolution, the role of international organizations in conflict resolution, example ECOWAS, African Union, United Nations.

GST223: Introduction to Entrepreneurship: (2 units)

Introductory Entrepreneurial skills: Relevant Concepts: Enterprise, Entrepreneur, Entrepreneurship, Business, Innovation, Creativity, Enterprising, and Entrepreneurial Attitude and Behaviour. History of Entrepreneurship in Nigeria. The rationale for Entrepreneurship,

Creativity, and Innovation for Entrepreneurs. Leadership and Entrepreneurial Skills for coping with challenges. Unit Operations and Time Management. Creativity and Innovation for • Self-Employment in Nigeria. Overcoming Job Creation Challenges. Opportunities for Entrepreneurship, Forms of Businesses, Staffing, Marketing, and the New Enterprise. Feasibility Studies and Starting a New Business. Determining Capital Requirement and Raising Capital. Financial Planning and Management. Legal Issues, Insurance, and Environmental Considerations.

GST223: Social Teaching of The Church: (0 unit)

General introduction; The nature of social doctrine; Historical dimension of social doctrine; General ethical principles and values; Biblical foundation of social teachings of the church; The church and promotion of social teachings: JDPC; Christian conception of the human person; Christian conception of human rights; Christian concept of social economy; Christian concept of human work/ private property; The lay faithful and social doctrine; The promotion of justice and peace /question of the just war.

200 LEVEL SECOND SEMESTER

SEN202: Principles of Programming Il (3 units)

Lecture and Tutorial Components: Builds on SEN102 advanced concepts of object-oriented programming, language implementation models, and development with a suitable object-oriented language. Topics include input/output, networking, threading, and GUI programming. Ample programming labs and projects form part of this course.

Practical Component (Use of Object-oriented programming to code Graphical User Interfaces, create applications, and engineer enterprise systems). Suggested Language: Java, Python, C#, C++

SEN204: Software Construction (2 units)

Lecture and Tutorial Components: Topics include specifications, abstraction techniques including typing, access control, inheritance, polymorphism, genericity, and design patterns. frameworks and architectures. Students will also learn the proper engineering use of techniques such as information hiding, classes, objects, inheritance, design by contract, exception handling, event-based systems, and concurrency. Special emphasis should be placed on suitable and appropriate object-oriented software development such as software quality and corresponding concepts, principles, and best practices for addressing both functional and non-functional requirements of the software system in its architecture. Reverse Engineering.

Practical Component (Use of an OOP language to show the core OOP properties – abstraction, encapsulation, inheritance, and polymorphism. Apply these properties in developing and engineering start-of-the-art industry-level applications). Suggested Language: Java, Python, PHP, Node.js, C#, C++.

SEN206: Design and Analysis of Computer Algorithm (2 units)

Lecture and Tutorial Components: Introduction to algorithms and their importance, mathematical foundations: growth functions, complexity analysis of algorithms, summations, recurrences, sorting algorithms. Algorithm design: divide-and-conquer approach, greedy approach. Graph algorithms: graph searching, topological sort, minimum spanning tree, shortest paths, backtracking, and its applications in games. String matching. Dynamic programming and longest common subsequence. Theory of NP-completeness. Turing machines and the halting problem.

Practical Component: (Solving practical problems using algorithms and pseudocodes; Applying the algorithms and pseudocodes in developing practical software solutions and applications). Suggested Language: Java, Python, C, C#, C++, Node.js, PHP

SEN208: Principles of Operating Systems (2 units)

Lecture and Tutorial Components: Surveys methods and algorithms used in operating systems. Concurrent distributed operation is emphasized. The main topics covered are an introduction to operating systems, process management, process scheduling, inter-process communications, memory management techniques, virtual memory, I/O management, deadlock avoidance, file system design, socket programming, distributed operation; distributed data; performance evaluation, protection, and security. Server operating system: for both local and cloud servers.

Practical Component: (Installation/server roles, remote desktop management, windows firewall, active directory/group policy, file system, DNS, DHCP, backup management: for both local and cloud servers). Suggested Tools: Windows Server 2019; Linux Server; VMware.

SEN210: Software Engineering Process (2 units)

Lecture and Tutorial Components: Software process definition — software process management and infrastructure, Software life cycles — categories of software processes, software life cycle models, software process adaptation, practical considerations; Software process assessment and improvement software process assessment methods, software process improvement models, and continuous and staged software process rating; Software measurement software process and product measurement, quality of measurement results, and software process measurement techniques; Software engineering process tools. CASE tools. Software methodologies (e.g., Agile: Scrum, eXtreme programming, model-driven development methodology, test-driven development, Kanban, Lean, feature-driven development, dynamic software development methodology, crystal methodologies, behavior-driven development, example-driven methodology, domain engineering approach, etc.), Agile methodologies and processes, DevOps methodologies and processes.

Practical Component (Practically using DevOps and Agile methods and other lean methods in software development projects; Building software products using agile methods). Suggested tools: Jira, Confluence, Enterprise Architect, Python, Java, PHP, Node.js, C#, C++

SEN212: Information Visualization (2 units)

Lecture and Tutorial Components: Introduction to data and information visualization; History of information visualization; Visual perception, color, and narrative; Cognition and visual perception, the aesthetics of visual media; Theoretical, practical, and aesthetic perspectives on information visualization; Visual representations; Visual mapping; Techniques for processing and manipulating information for the purpose of visualization; Temporal & statistical visualization: studies of spatial, relational, multivariate, time-series, interactive, and other visual approaches; Design and narrative for visualization; Mapping, counter-mapping, and geospatial visualization; Network visualization; Evaluation of information visualization; Meta-modelling.Use of visualization software tools on data and information: Developing related applications

Practical Component (Developing information visualization applications, design, and implementation of static and dynamic graphs, dashboards, tables, charts, images, maps, spreadsheets, visual presentations, reports, complex UIs, etc; Developing information visualization applications). Suggested Tools: Tableau, R, MATLAB, Grafana, Microsoft PowerBI, Scimago Graphica, Python/Django/Flask, Java, C#, C++, Node.js, PHP/Laravel.

SEN282: Software Engineering Lab IV (2 units)
(Games Applications Development Projects)

Creation of basic games – Ludo, Cards, Checkers, Ayo, Chess, Scrabble, Monopoly, Bubble, Quizzes. Development of Gaming characters, Event-based interactions between characters, Animations and Transitions, Effects, and Media (image, sound, and video). Using Blender for Cartoon production. Using Unity 3D, Cinema 4D, Python, C#, C++, and Java for game app development.

SEN290: Students Industrial Work Experience Scheme (SIWES) I (3 units)

This is the first part of the SIWES program. Students are to be attached to software firms for a period of three months (12 weeks) with a view to making them acquire practical experience and to the extent possible, develop skills in all areas of software engineering computing. Students are supervised during the training period and shall be expected to keep records designed for the purpose of monitoring their performance. The SIWES will be undertaken during the three months long vacation. Students are also expected to submit a report on the experience gained and defend their reports. In addition, during the internship, students are to develop and engineer a completely innovative and creative application which they will also defend immediately after the SIWES program. The assessment for this course is as follows: Software Project (50 marks), Logbook (20 marks), SIWES I Report (15 marks), Industry Supervisor's Report (15 marks).

300 LEVELFIRST SEMESTER

SEN301: Object-Oriented Analysis and Design (2 units)

Lecture and Tutorial Components: Object-oriented approach to information system development, particularly in reference to the earlier stages of analysis and design. Importance of modeling, principles of modeling, object-oriented modeling, a conceptual model of the Unified

Modelling Language (UML), architecture, and software development life cycle. The principles and basic concepts of object orientation and the different aspects of object-oriented modeling as represented by the UML technique. Case study of a typical UML—based CASE tool. Object-oriented modeling using UML with pseudocodes: use case diagram, misuse case diagram, activity diagram, sequence diagram, collaboration diagram, state machine diagram, class diagram, etc., also, entity relationship diagram, data flow diagram, etc.

Practical Component (Design modeling (design phase UML modeling): Object-oriented modeling using UML with pseudocodes: use case diagram, misuse case diagram, activity diagram, sequence diagram, collaboration diagram, state machine diagram, class diagram, etc., also, entity relationship diagram, data flow diagram, etc.). Suggested Tool: StarUML, E-Draw, ConceptDrawMindMap, Adobe Illustrator, Enterprise Architect, etc.

SEN303 Software Testing & Quality Assurance (2 units)

Lecture and Tutorial Components: How to assure it and verify it, and the need for a culture of quality. Avoidance of errors and other quality problems. Inspections and reviews. Testing, verification, and validation techniques. Process assurance vs. Product assurance. Quality process standards. Product and process assurance. Problem analysis and reporting. Statistical approaches to quality control. White box and black box testing. Bottom-up and top-down testing. The test case design, Reviews expert opinions, formal design reviews, peer reviews, and team reviews. Acceptance tests. SQA defect removal effectiveness, cost, and metrics. White box testing: Use of stubs and drivers for incremental testing, correctness tests, and line coverage, McCabe's Cyclomatic Complexity metrics; Black box testing: test cases and boundary values; documentation tests, reliability tests, and stress tests. Other tests: maintainability tests, flexibility tests, testability tests, portability tests, reusability tests, and interoperability tests. Software Testing implementation: unit tests, integration tests, system tests. Automated testing, load tests. Alpha and beta testing; CASE tools for SQA.

Practical Component (Developing test cases and carrying out testing and reviews on Software applications, misuse case scenarios, practical software verification, and validation; Defect analysis; test documentation, traceability matrices; testing estimation techniques). Suggested Tools: Enterprise Architecture, Telerik Test Studio, Selenium.

SEN305 Web Application Development (3 units)

Lecture and Tutorial Components: Covers client-server model for web applications and associated client-side and server-side technologies, MVC development guidelines, and development of a complete web application using a framework such as Ruby on rails or Django.

Practical Component (Developing enterprise applications using Model View Controller and Model View Template paradigms, Writing Web APIs and libraries). Suggested Tool: Python/ Django/Flask, PHP/Laravel, C#.Net, ASP.Net, Node.js, MySQL database.

SEN307 Database Systems (3 units)

Lecture and Tutorial Components: The course will cover the concept, principles, components, development, and application of database systems. The conceptual models and

structures necessary for designing and implementing a relational database system will be taught. Topics to be covered: entity-relationship, relational data models, relational algebra, SQL, normalization, file organization, indexing, hashing, and enterprise-wide web-based applications that employ databases. Rational Databases: Mapping conceptual schema to relational Schema; Database Query Languages (SQL) Concept of Functional dependencies &multi-Valued dependencies Transaction processing; Distributed databases; Relational Algebra; SQL; PL/SQL; Relational Calculus; Functional Dependencies & Normal Forms; Design Algorithms; Database System Access and Storage: Disk, Buffer, and File Management; Indexing Technique; Transaction Management. Database designs, normalization, practical database implementation using MySQL, Microsoft Server, etc

Practical Component (Database designs, normalization, practical database implementation using MySQL, Microsoft Server, etc. Developing Artificial Intelligence applications using Transact SQL). Suggested Tool: Microsoft SQL Server, MySQL, Python, C#, C++

SEN309: Concepts of Programming Languages (3 unit)

Lecture and Tutorial Components: Preliminaries, the evolution of programming languages, paradigms, language design considerations, language processing including syntax and semantic analysis, naming, binding, type checking, expression and assignment statement, statement-level control structures, subprograms, abstract data types, support for object-oriented languages, concurrency, exception handling, functional and logic programming.

Practical Component (Implementing the above concepts using at least two languages or paradigms). Suggested Languages: Python, Java, Node.js, PHP, C#, C++

SEN381: Software Engineering Lab V (2 units)
(Blockchain Projects)

Applying Algorithms, Hashing, Creating Cryptocurrencies, Creating Blockchain ledgers, creating blockchain transactions, developing blockchain-based web and mobile apps, developing wallets and firewalls. Implementing Smart contracts and Decentralized Applications. Implementing trust, NFTs.

GST311: Introduction to Entrepreneurship Studies: (2 Units)

Profiles of business ventures in the various business sectors such as:

Soap/Detergent, Toothbrush and Toothpaste making; Photography; Brick making, Rope making; Brewing; Glassware production/Cera1nic production, Paper production; Water treatment/conditioning/packaging; Food processing/preservation/packability; Metal fabrication; Tanning industry; Vegetable oil extraction: Fanning; Fisheries/aquaculture; Plastic making; Refrigeration/Air-conditioning, Carving, Weaving; Bakery; Tailoring; Printing; Carpentry; Interior Decoration; Animal husbandry, etc. Case Study Methodology applied to the development and administration of Cases that bring out key issues of the business environment, start-up, pains and gains of growth of businesses, etc. with particular reference to Nigerian businesses. Experience sharing by business actors in the economy with students during-Case presentations.

S'I'A343: Operations Research (3 units)

Nature and scope of operations research. Linear programming and graphical, simplex (including' big M and two-phase) methods. Sensitivity analysis. Duality theory. Transportation and assignment problems. Network analysis: CPM and PERT. Inventory theory and applications. Sequencing and scheduling.

300 LEVEL SECOND SEMESTER

SEN302: Software Configuration Management & Maintenance (2 units)

Lecture and Tutorial Components: Management of the software configuration management process — organization context for software configuration management, constraints and guidance for software configuration management process, planning for software configuration management, software configuration management plan, and surveillance of software configuration management, Software configuration identification, and software library; Software configuration control requesting, evaluating and approving software changes, implementing software changes, and deviations and waivers: Software configuration status accounting — software configuration status information and reporting; Software configuration auditing. Key issues in software maintenance technical issues, management issues, maintenance cost estimation, software maintenance measurement, Maintenance process maintenance processes, and activities; Techniques for maintenance program comprehension, reengineering, reverse engineering, migration, and retirement.

Practical Component (Implementing Version Control, Software Project Management and carrying out updates, bug tracking, etc). Suggested Tools: Git Version Control, Jira and Confluence, Enterprise Architect.

SEN304: Software Engineering Project Management (2 units)

Lecture and Tutorial Components: What is project management, the role of a project manager, product lifecycle, product management process, problem template, assumption mapping, market validation, minimal viable product (MVP) design, user stories design, feature prioritization with MOSCOW method, effort/value mapping, t-shirt sizing method, product report portfolio design. Determination and negotiation of requirements, feasibility analysis, and process for the review and revision of requirements, Software project planning process planning, determine deliverables, effort, schedule and cost estimation, resource allocation, risk management, quality management, and plan management, Software project enactment implementation plans, software acquisition and supplier contract management, implementation of the measurement process, monitor process, control process, and reporting; Review and evaluation determining satisfaction of requirements, and reviewing and evaluations performance, Closure determining closure and closure activities: Software engineering measurement establish and sustain measurement commitment, plan the measurement process, perform the measurement process, and evaluate measurement, Software engineering management tools.

Practical Component (Use of project management tools to manage the Software Engineering processes and projects using principles such as agile, etc). Suggested Tools: Jira and

Confluence, Enterprise Architect, Asana for project management, Slack for communication and file sharing, etc.

SEN306: Research Methodology (I unit)

Foundations of research; problem identification and formulation; research design; qualitative and quantitative research; measurement: sampling; data analysis; Interpretation of data and paper writing; use of encyclopedias, research guides, handbooks, etc., academic databases for the computer science discipline; use of tools/techniques for research: reference management software, software for detection of plagiarism.

SEN308: Engineering Professional Practice (2 units)

Lecture and Tutorial Components: Accreditation, certification and licensing. codes of ethics and professional conduct, nature and role of professional societies and software engineering standards, and economic impact of software. Employment contracts, legal issues, documentation, and trade-off analysis: Group dynamics and psychological dynamics of working in teams/groups, individual cognition, dealing with problem complexity, interacting with stakeholders, dealing with uncertainty, and reading. understanding and summarizing writing, team and group communication, and presentation skills. Software engineering as a business. Software engineering entrepreneurship; Softwarepreneurship.

Practical Component(Practical development of software products in pairs/teams/groups while applying SE codes of ethics, professional conduct, and SE standards; Communication and presentation skills; Drafting of contracts; Bidding for contracts/jobs; Contract negotiations & costing; Legal issues in SE; Employment contracts; Marketing software products; SE as a business; Thinking and acting as a Business man/woman, other software engineering professional practice; Managing software start-ups; Managing software company; Typifying and demonstrating a software firm in practice; Practical role-playing of software industry scenarios. Software engineering entrepreneurship, Students are encouraged to set up and register software companies and businesses. Assessments for this practical component will be via presentations, oral interviews and role play.

SEN310: Software Engineering Security (2 units)

Lecture and Tutorial Components: History and terminology, security mindset, design principles, system/security life cycle, security implementation mechanisms, in the assurance analysis model, disaster recovery, and forensics; Security mechanism cryptography, authentication, redundancy, and Intrusion detection: Operational issues trends, auditing, cost/benefit analysis, asset management. standards, enforcement, legal issues, and disaster recovery; Policy creation policies, maintenance of policies, prevention, avoidance, incident response (forensics), and domain integration (physical, network, internet, etc.); Attacks social engineering, denial of service, protocol attacks, active and passive attacks, buffer overflow attacks, and malware, Security domains security awareness and possible domains, forensics legal systems, digital forensics and its relationship to other (forensic disciplines, rules of evidence, search and seizure, digital evidence, and media analysis; Security services; Threat analysis model; Vulnerabilities; Application security.

Practical Component (Developing practical applications related to software-defined networking & security, cyber security, and other software security domains. Development of software security applications, antivirus applications, virtual private network apps, firewalls, engineering innovative secured applications using improved and state-of-the-art security algorithms, engineering secured apps): Suggested Tools: Python, Java, C#, C++, Node.js, etc.

SEN312: Al and Expert Systems (2 units)

Lecture and Tutorial Components: The evolution of computing, defining artificial intelligence, general problem solving: approaches in artificial intelligence, characteristic requirements for the realization of intelligent systems, programming languages for artificial intelligence, and architecture for artificial intelligence machines. The psychological perspective of cognition; production: systems; problem-solving by intelligent search; the logic proposition and predicates; default and non-monotonic reasoning. A structural approach to knowledge representation, the nature, and goals of soft computing. The nature of expert systems; types of applications of expelling systems; the relationship of expert systems to artificial intelligence and to knowledge—basic systems; distinguishing features of expert systems; theoretical foundations; basic forms c inference; the representation and manipulation of knowledge in a computer; rule-base representations; logic-based representations; frames; semantic and partitioned nets; basic components of an expert system; generation of explanations; handling of uncertainties; truth maintenance systems; expert system architectures; an analysis of some classic expert systems; building expert systems; methodologies for building expert systems: knowledge acquisition and elicitation, formalization, representation, and evaluation; knowledge: engineering tools; expert systems paradigms.

Practical Component (Developing and engineering AI and expert systems applications: Implementing problems that involve Classification, Regression, and Clustering; Building emotionally sensitive applications and other intelligent systems; Implementing AI on web, mobile, networks, games, database, wearable devices and machine platforms and systems, etc.). Suggested Tools: Microsoft Kinect 2.0, Python, C++, etc.

SEN314: Engineering Mobile Applications (2 units)

Lecture and Tutorial Components: Introduction to developing mobile applications, beginning with mobile operating system capabilities and application architecture and extending to major components, such as activities, services, broadcast receivers, etc. Development of interactive applications using widget libraries, web-based services, animation, an SQL database engine, and multithreading.

Practical Component (Developing Mobile Applications such as Utility and Application software, scaling them for different mobile devices, architectures and operating systems). Suggested Tools: Xamarin, Flutter, Java, Python, C#, C++, Django, React native.

SEN316: Embedded Systems (2 units)

Lecture and Tutorial Components: Introduction to embedded computing and embedded systems; typical embedded systems _core of the embedded system, memory, communication

interface; embedded firmware; embedded real-time operating systems; real-time operating systems—based embedded system design; task communications and synchronization.

Practical Component (Building applications that infuse embedded systems). Suggested Tools: Micro Python, C#, C++, TinkerCAD, Proteus & Wokwi Simulators.

SEN382: Software Engineering Lab VI (2 units)
(Embedded Systems Projects)

Creating Motor and cruise control systems, Implementing multimedia in cars. Incorporating embedded systems in home appliances (Television, Microwave, Refrigerator, Fan, Doors, Air Conditioner, Washing Machine, Toasting Machine, Blending Machine, Lamps, Security Cameras, Home Electronic devices, etc.), Traffic Control systems, Implementing Vehicle Tracking systems, and Temperature control systems. Electronic Billing Meters. Building drones; Internet of Things applications; Software Defined Communication Systems, Embedded Systems in Wearable devices, Electronic billboards, Public Address Systems, Network Systems, Automobiles, Smart Classrooms, Offices, Farms, Security devices, Manufacturing Industries, Electrical and Electronic devices, smart wristwatches, smart rings, smart eye glasses, autonomous vehicles, etc.

SEN390: Students Industrial Work Experience Scheme (SIWES) II (3 units)

This is the second part of the SIWES program. Students are attached to software firms for a period of three months (12 weeks) with a view to making them acquire practical experience and to the extent possible, develop skills in all areas of software engineering computing. Students are supervised during the training period and shall be expected to keep records designed for the purpose of monitoring their performance. The SIWES will be undertaken during the three months long vacation. Students are also expected to submit a report on the experience gained and defend their reports. In addition, during the internship, students are to develop and engineer a completely innovative and creative application which they will also defend immediately after the SIWES program. The assessment for this course is as follows: Software Project (50 marks), Logbook (20 marks), SIWES II Report (15 marks), Industry Supervisor's Report (15 marks).

400 LEVEL FIRST SEMESTER

SEN401: Software Engineering Economics (2 units)

Lecture and Tutorial Components: Software engineering economics fundamentals, lifecycle economics: Risk and uncertainty goals. estimates and plans, estimation techniques, addressing uncertainty, prioritization, decisions under risk and uncertainty, Economic analysis methods for-profit decision, cost-benefit analysis, cost-effectiveness analysis, break-even analysis, business case, multiple attribute evaluation, and optimization analysis: Practical considerations — the "good enough" principle, friction-free economy, ecosystems, and offshoring and outsourcing. Development of effort estimation; Performance Models, Optimal Performance, Sensitivity Analysis, Cost-Effectiveness Models; Software cost – estimation: COCOMO.

Practical Component (Risk analysis; cost-benefit analysis; cost-effectiveness analysis; break-even analysis; optimization analysis; effort estimation; cost estimation; COCOMO I; COCOMO II, etc.

SEN403: Human Computer Interaction (3 units)

Lecture and Tutorial Components: Introduces the principles of user interface development, focusing on three key areas: (l) Design: How to design good user interfaces, starting with human capabilities and using those capabilities to drive design techniques: task analysis, user-centered design, iterative design, usability guidelines, interaction styles, and graphic design principles. (2). Implementation: Techniques for building user interfaces, including low-fidelity prototypes, Wizard of Oz. and other prototyping tools; input models, output models, model-view-controller, layout. , constraints, and toolkits. (3). Evaluation: Techniques for evaluating and measuring interface usability, including heuristic evaluation, predictive evaluation, and user testing. Interaction design principles, user-centered design (UCD) principles, activity theory, task analysis, design thinking, participatory design, co-design, and empathy design.

Practical Component (Building Professional Human-Computer Interfaces using HCI principles: User story, user scenario, user story map, user journey, user personas, user flow diagrams, user storyboard prototyping, low-fidelity wireframes, and high-fidelity wireframes and wire flows. Practical UX/UI design and implementations; Development of simple applications with good human-computer interaction qualities). Suggested Tools: Adobe XD, Sketch, Figma, HTML, CSS, JavaScript, PHP, C#, C++, and Python: Required Materials: drawing book, sticky note pack (with multiple colors), cardboard sheets.

SEN405: Open-Source Software Development & Applications (2 unit)

Lecture and Tutorial Components: Introduces concepts, principles, and applications of open-source software. Discusses the open-source software development process. Covers economy, business, societal, and intellectual property aspects of open-source software. Obtain hands-on experience on open-source software and related tools through developing various open-source software applications such as mobile applications and Web applications building on existing open-source frameworks and application development platforms.

Practical Component (Building and Deploying Mobile and Web applications on Open-source platforms, contributing and collaborating on these platforms). Suggested Tools: Flutter or Xamarin, Django Framework, Git Version Control.

SEN407: Distributed, Parallel and Cloud Computing (2 units)

Lecture and Tutorial Components: Analysis and Design of Parallel and Distributed Algorithms: Languages Operating Systems for parallel processing: GPGPU computing, Architecture of parallel distributed systems. Tools for parallel computing. Parallel (distributed) database system. Networking aspects of parallel/distributed computing, Parallel/distributed scientific computing Applications; high-performance computing Applications in molecular sciences: Multimedia applications for parallel/distributed systems; Grid networks, services, and applications; Distributed File System, Hyper-Scale/Hyper-Converged Distributed Storage Design, Storage I/O Protocols: Cloud as a Service, Cloud Infrastructure, Management and

operations, Performance, Scalability, Reliability, Virtualisation, loud Provisioning Orchestration, Architecture support, development tools, Platforms and Applications, Legal aspects and Service Level Agreement, Mobile computing advances in the Cloud, Performance optimization

Practical Component (Building and deploying distributed, parallel, and cloud-based software applications): Suggested Tools: PHP/Laravel, Python/Django/Flask, Node.js, Java, C#, C++

SEN409: Software Architecture and Design (2 units)

Lecture and Tutorial Components: An in-depth look at software design. Continuation of the study of design patterns, frameworks, and architectures. Survey of current middleware architectures. l) designing distributed systems using middleware. Component-based design. Measurement theory and appropriate use or metrics in design. Designing for qualities such as reliability, performance, safety, security, reusability, etc. Measuring internal qualities and complexity of software. evaluation and evolution of designs. Basics of software evolution, reengineering, and reverse.

Practical Component (Practical software architectural designs, Design patterns, frameworks, and architecture; distributed systems designs; component-based designs; designing for quality, measuring software quality and complexities; software reengineering, forward and reverse engineering: Applying software architecture and design principles in the software engineering process). Suggested Tools: Drawing books, pencils, rulers, EDraw, Adobe XD, etc.

SEN411: Special Topics in Software Engineering (2 units)

Recent topics and developments in software engineering are expected to be introduced from year to year. Apart from seminars to be delivered by lecturers or guests, students are expected to do substantial readings on their own.

Elective 1: Software Defined Networks

Lecture and Tutorial Components: Computer networks and networking, Software defined networks: Introduction to programmable networks, History of Software Defined Networking (SDN) – Modern Data Center – Traditional Switch Architecture – Why SDN – Evolution of SDN – How SDN Works – Centralized and Distributed Control and Date Planes; Open Flow Specification – Drawbacks of Open SDN, SDN via APIs, SDN via Hypervisor- Based Overlays – SDN via Opening up the Device – SDN Controllers – General Concept; Multitenant and Virtualized Multitenant Data Center – SDN Solutions for the Data Center Network – VLANs – EVPN – VxLAN – NVGRE; Programming SDNs: Northbound Application Programming Interface, Current Languages and Tools, Composition of SDNs – Network Functions Virtualization (NFV) and Software Defined Networks: Concepts, Implementation and Applications; Juniper SDN Framework – IETF SDN Framework – Open Daylight Controller – Floodlight Controller – Bandwidth Calendaring – Data Center Orchestration.

Practical Component (Designing, simulating, and building networks; Programming networks, Developing applications for networks, Building software-defined networks): Suggested Tools: Python, Java, C++, C, C#

Elective 2: Artificial Intelligence, Machine Learning, and Deep Learning with TensorFlow

Lecture and Tutorial Components: Understanding artificial intelligence, introduction to TensorFlow, architecture of TensorFlow, installations for TensorFlow: Keras, TensorFlow, Python, R Studios, etc., mathematical foundations, artificial neural networks (ANN) in TensorFlow, convolutional neural networks (CNN) in TensorFlow, recurrent neural networks (RNN) in TensorFlow, supervised, unsupervised (self-supervised) and semi-supervised deep learning; attention branch network (ABN), contrastive learning. TensorBoard visualization, TensorFlow word embedding, single layer perceptron, TensorFlow linear regression, TFLearn and its installation, CNN & RNN difference, TensorFlow - Keras, TensorFlow - Distributed computing, TensorFlow - Exporting, multi-layer perceptron learning, hidden layers of perceptron, TensorFlow – Optimizers, TensorFlow – XOR implementation, Gradient descent optimization, TensorFlow – forming graphs, object detection, image recognition using TensorFlow, audio recognition, TensorFlow debugging and fixing problems, style transferring, TensorFlow APIs, TensorFlow security, TensorFlow single and multiple GPU, TensorFlow mobile.

Practical Component (Set up TensorFlow, load dataset, build machine learning/ deep learning models, Train and evaluate the models; Building complex AI, ML & DL applications with Keras and TensorFlow: computer vision, natural language processing, text processing, voice processing, speech recognition, voice recognition, image processing, image recognition, fraud detection & prevention, video processing, video detection, pattern recognition, motion detection, facial detection, biometric detection, remote sensing, human activity recognition, human pose recognition, attitude recognition, action recognition, attention mapping, gesture recognition, motion recognition, behaviour recognition, posture recognition, cybersecurity, predictive analytics, human action recognition, behaviour analysis, feature extraction, times series data, robotic, augmented reality, animation and gaming, point cloud analysis, etc. Implementing AI on web, mobile, networks, games, database, wearable devices and machine systems, etc.): Suggested Tools: Python, Keras, TensorFlow, C++, JavaScript, etc.

SEN413: Research Seminar (1 unit)

Research seminar in software engineering fields. The seminar paper must be novel, relevant, and timely and must address theoretical and practical gaps in the literature. There will be no written examination for the course, however, students will be assessed based on their novel seminar papers. All seminar papers must be presented and assessed by a team of assessors.

SEN415: Visual Design (2 Units)

Lecture and Tutorial Components: Introduction to Visual Design. Design Theory; Information Design, Color, Visual Perception Theory. Color symbolism, meaning, and cultural variation; Color Theory; Typography and Typographical Elements; the Design Process; Symbolism; Collage: Collage; Photomontage; Assemblage; Digital collage/e-Collage; Influence of movements. Visual Identity and Branding: Visual branding; Visual identity; Logo design. UI, UX, and Design for the Web.Model-driven development (MDD)

Practical Component (Design and implementation of graphics, innovative interfaces, industrial products, emojis, animations, complex GUIs, etc. Creating Visual Brand Identity Guides, Implementing Branding elements, developing UI / UX applications; Developing applications with good visual design). Suggested Tools: Adobe XD, EDraw, Figma, Adobe Illustrator, Blender, Unity; Tableau, R, MATLAB, Grafana, Microsoft PowerBI, Scimago Graphica, Python, C#, C++.

SEN481: Software Engineering Lab VII (2 units)
(Robotics Projects)

Developing full-fledged robotic applications (software robots and software-driven hardware robots) – Computer Vision (Eyeglasses for the blind), Intelligent chat systems. Applying manipulators, Kinematics, and Tactile Sensors, Building drones, Service delivery robots. Virtual reality games and products. Image processing and recognition. Applying Neural Networks. Deep reinforcement learning in robotics.

400 LEVEL SECOND SEMESTER

SEN402: Enterprise Application Development (2 unit)

Lecture and Tutorial Components: The Enterprise, Enterprise Systems, and ERP, Web Applications, Enterprise Systems Architectures, Data-driven Web Applications, Managing an ERP, Specialty Enterprise systems; Enterprise Application Development: Enterprise applications, Java EE technology, Simplifying Java EE, Design patterns; Spring Framework Fundamentals: Spring Architecture, Introducing IoC, Spring IoC, Applying IoC to Rainforest; The Spring JDBC Framework: Evaluating JDBC, JDBC Template, Applying Spring JDBC to Rainforest; Aspect-Oriented Programming (AOP): AOP with Spring, Schema-based AOP, AspectJ Pointcut Expression, Annotation-based AOP; Transaction Management: Java EE Transaction, Motivation and Architecture of Spring Transaction Support, Declarative Transaction Control; Spring MVC: Spring MVC overview, Spring MVC Component Architecture, Configuring Spring MVC, Spring MVC Development Process, REST-Style URLs, AJAX Controllers; Processing Forms and Validation: Form processing with Spring MVC; Introduction to Hibernate: Object-Relational Mapping, Hibernate Architecture, Working with Hibernate; Mapping Persistent 20 Classes: One-to-one relationships, Strategies for Handling Inheritance, One-to-Many Relationships; Working with Persistent Classes: Hibernate Architecture and Object Persistence, Hibernate Query Language, Executing Native SQL; Hibernate Performance and Spring Integration: The Hibernate Caching Architecture, Integrating Hibernate with Spring.

Practical Component (Building SOAP and Restful Web API and interfacing applications built on different languages with the APIs; Building enterprise applications). Suggested Tools: Java, C#, C++, Python, PHP, Node.js, etc.

SEN404: Virtual and Augmented Reality (2 units)

Lecture and Tutorial Components: Introduction to Virtual Reality-fundamental concept and components of virtual reality, primary features and present development on virtual reality;

Multiple modal interactions: multiple modals of input and output interface in Virtual Reality-Input—Tracker, sensor, digital clove, movement capture, video-based 3D Scanner, etc., Output—Visual/auditory/haptic devices, visual auditory-haptic, interaction immersion and imagination, Visual computation and environmental modeling: Visual computation in virtual reality—Fundamentals of computer graphics; real-time rendering technology, principles of stereoscopic display, software and hardware technology on stereoscopic display; Environmental modeling in virtual reality: Geometric modeling (geometric behavior), behavior stimulation, and physically based simulation; management of large scale environment, augmented reality (AR), mixed reality (MR), digital entertainment, development and applications of VR. AR, and MR; Haptic and Force Interaction in Virtual Reality: Concept of haptic interaction, principles of touch feedback and force feedback, typical structure, and principles of touch/force feedback facilities in an application; Augmented reality: System structure of Augmented Reality, Key technology in AR, general solution for calculating geometric & illumination consistency 21 in the augmented environment; VR development tools: Frameworks of software development tools in VR, Modeling tools for VR, X3D Standard, Vega, MultiGen, Virtools, etc.; Applications of VR in Digital Entertainment: VR technology in Film & TV production, VR technology in physical exercises and games, Demonstration of digital entertainment by VR; Application of VR in other domains.

Practical Component (Simulating VR, AR, and Mixed Reality for everyday situations such as conference calling, educational settings, games and storytelling, etc.; 3D Designs and animations; Building VR, AR and MR applications). Suggested Tools: Autodesk's 3D visual prototyping and visualization software, VRED with Python, OpenGL, Java, C#, C++.

SEN406: Fault—Tolerant Computing (2 units)

Introduction and overview of fault-tolerant schemes; fault and error modeling; t generation and fault simulation; concepts in fault-tolerance; reliability/availability model system-level diagnosis; low-level fault-tolerance — coding techniques (basic principles, pa: bit codes, hamming codes, error detection and retransmission codes, burst error correct codes, Reed-Solomon codes, etc.); high-level fault-tolerant techniques in the system: rollback checkpointing, reconfiguration; software fault-tolerance; fault-tolerant routing; Integra hardware/software fault-tolerance; redundancy, spares and repairs — apportment, system versus component redundancy, parallel redundancy, RAID system reliability, N-modular redundancy; software reliability and recovery techniques, network system reliability optimization.

SEN408: Game Design and Development (2 units)

Lecture and Tutorial Components: The course covers game development history, platforms, goals and genres, player elements, story and character development, gameplay, levels, interface, audio, development team roles, game development process, and marketing and maintenance. 3D designs and animations; Students will design playful games, analyze them, and complete portions of animated game designs with appropriate documentation.

Practical Component (3D games designs and animations; Cartoon animation and production, Design and develop characters, transitions, interfaces and audio for games; Building games applications). Suggested Tools: Unity with Java and Python, Blender, C#, C++.

SEN410: Modelling and Computer Simulation (2 units)

Lecture and Tutorial Components: Introduction to simulation concepts, introduction to models, problem formulation, project planning, system definition, input data collection and analysis, modeling translation, verification, validation, experimental design, analysis, project reports and presentations, and training simulators.

Practical Component (Carrying out modeling and simulation for various software designs and scenarios; developing software systems and applications). Suggested Tools: MATLAB, Visio, Edraw, Simulink, Python, Java, C#, C++.

SEN482: Software Engineering Lab VIII (2 units)
(DevOps Projects)

Applying Version Control with Git, Applying Continuous Integration and Continuous Deployment to projects, Ecosystem, and Networking, working with Kubernetes, Testing, Cloud Computing, working with the Command Shell, Working with Docker.

SEN490: Project (6 units)

An independent investigation of appropriate software, hardware, communication, and networks or IT-related problems in Software Engineering carried out under the supervision of a lecturer. The project should be geared at software development. Before registering, the student must submit a written proposal to the supervisor to review. The proposal should give a brief outline of the project, an estimated schedule of completion, and computer resources needed. A formal written report is essential and an oral presentation may also require. All projects must be software intensive, research-oriented, and professionally engineered (i.e., must be professional, creative, novel, relevant, and timely). Each project should engineer, produce and deliver a full, novel, and finished product using software engineering best practices.

21. Definition Of Concepts

21.1 Academic/Study Programme

An academic or study program is a combination of courses prescribed by the Department to be taken in a progressive manner from 100 level to 400 level and must be completed in a minimum of eight semesters (4 years) and a maximum of twelve semesters (6 years) leading to the award of a recognized qualification.

21.2 Courses

A course is a series of lectures on a particular subject as may be determined by the Department/Department, lasting one semester and associated with a single or two examination paper(s). A course may last for more than one semester, provided that such a course is divided into two parts, either part of which carries equal credit weighting and lasts for only one semester and is examinable at the end of the semester.

21.2.1 A Compulsory/Core Course

This is a course that must be taken and passed by the student before graduation or such a student is deemed to have fully satisfied the conditions for successful completion of the program. It may be a Departmental or borrowed course.

21.2.3 Elective Course

This is a course a student must select from the list recommended by the Department to be taken and passed for successful completion of the programme. A student has the liberty to make a choice provided the choice is within the Departmental recommended list.

21.3 Department

This refers to any teaching or research unit, institute, or any unit recognized by the University Council on the recommendation of the Senate.

21.4 Faculty

This refers to a unit as constituted and established by the Senate and Council of the University usually consisting of two or more Departments.

21.5 Dean of Faculty

Refers to a person elected by the Department Board of Studies and approved by the Vice Chancellor to oversee the affairs of the Departments in the Faculty.

21.6 Head of Department

Head of Department (HOD) refers to the person appointed by the Vice Chancellor to oversee the affairs of the Department.

21.7 Academic Year/Session

This is a 9 month or 36-week period beginning in October of one year and ending in July of the next.

21.8 Semester

This is one half of an academic year; usually a duration of 18 weeks.

21.9 Year of Study

This refers to the year within the prescribed duration of study by which a student may be identified.

21.10 Academic Discipline

It refers to a special area of study normally domiciled in a department; for example, Software Engineering in the Department of Software Engineering.

21.11 Repeat Course/Examination

A repeat course/examination is a course/examination taken by a student who had failed a course and who must register and repeat lectures in a course after failing in the previous attempt(s).

21.12 Carry Over
A carry over course is a course which a student had registered for in a semester and failed and would have to register for it again in the following session.

21.13 Board of Examiners
This refers to all members of a department or department holding academic appointments above the rank of Assistant Lecturer and who taught courses.

21.14 Supplementary Examination
This is an examination taken by students who were unable to sit a particular examination on account of ill-health or any other accepted reasons. The examination must be taken at the next available opportunity. Such a student is credited with the full marks so earned.

21.15 Pass Grade
A grade within the range A-D

21.16 Fail Grade
A grade of E or F.

21.17 Concurrent Course
A course listed between two or more departments

21.18 Good Standing
This refers to a student whose CGPA is not less than 1.50 at the end of each year of study.

21.19 Probation
It is a period (session) in which a student whose CGPA at the end of the session falls below 1.50 is given to improve upon.

21.20 Withdrawn
It is the compulsory termination of studentship in a program when a student's CGPA falls below 1.50 after probation.

21.21 Withdrawal
This is the voluntary termination of studentship at the student's volition for whatever reason.

21.22 Grade Point (GP)
The Grade Point derives from the actual percentage raw score for a given course; the raw score is converted into a letter grade and a grade point

21.23 Grade Point Average (GPA)
This is the average of weighted grade points earned in the courses taken during the semester. The Grade Point Average is obtained by multiplying the grade point attained in each course by the number of Credit Units assigned to that course, and then summing these up and dividing by

the total number of Credit Units taken for the semester. The GPA is used to report the academic performance of each student in a semester.

21.24 Cumulative Grade Point Average (CGPA)

This is the up-to-date mean of the Grade Points earned by the student in a program of study. It is an indication of the student's overall academic performance at any point in the training program. To compute the cumulative grade point average, the total of the grade points multiplied by the respective Credit Units for all the semesters are added and then divided by the total number of credit units for all courses registered by the student.

21.25 Credit Units

Credit Units consist of a specified number of students – teacher contact hours per week per semester. Credit units are used in two complementary ways, one, as a measure of course weighting and the other, as an indicator of student workload.

22. General Examination Regulations

A. Requirements for taking an examination

In order to be admitted to a university examination, the student MUST:

- have duly registered for the course
- follow the approved course of study for a prescribed period
- have paid all prescribed fees by Senate as and when required
- satisfy 80% attendance at lectures
- Comply with any additional requirements approved by Senate from time to time.

B. Organization of Examination

- Each Faculty /Department shall be responsible for the organization and conduct of examinations for all courses taught in the Faculty/Department.
- Each Department shall appoint an examination officer who shall be responsible for preparing the timetable, arranging for invigilation, and other requirements necessary for conducting all examinations in the Department. The responsibility for supervision of examinations shall rest with the Head of Department as the Chief Examiner for the Department.
- The Sub-Dean of the Faculty shall be the Faculty Examination Officer. The Faculty Examinations Committee shall consist of each departmental examination officer with the Sub-Dean as Chair. The Committee shall be responsible for the coordination of timetables, the physical arrangement of examination venues, invigilation, and discipline in all examinations in the Faculty.

C. Scheduling of Examinations

Course examinations shall be scheduled at the end of the semester in which the teaching of the course is completed and on dates approved by Senate. If the University, for unavoidable reasons, is obliged to postpone an examination, the Registrar, in consultation with Deans of affected Faculties and the Director of Academic Planning shall re-schedule such examinations.

D. Duration of Examinations

- The duration of written examinations shall normally be between two-three hours depending on the Credit units of the course.
- The duration for practical examinations shall be a minimum of three hours.

E. Examination Timetable

The Director of Academic Planning shall compile and publish a draft timetable for all University examinations at least four weeks before the commencement of each examination and a final timetable not later than two weeks before the commencement of the examinations.

F. Setting and Administration of Examination Questions

- Each Lecturer teaching a course shall submit questions to the Head of the Department at least two to the examination date.
- The Chief Examiner will in turn arrange for the moderation of the questions.
- The Head of Department shall ensure that all results of the courses examined are collated by all the examiners of the courses and submitted along with answer scripts, attendance register, and marking scheme(s) two weeks after the examination as approved by Senate.
- The Head of Department shall be held responsible for any lapses arising from teaching, examination, preparation, and submission of results as and when due as approved by the Senate.

G. Invigilation of Examination

For each examination, there shall be a chief invigilator provided he/she is not the course lecturer of the said course. The course lecturer shall be available as an assistant examiner.

Duties of Invigilators

The chief invigilator shall

- Collect examination answer booklets and question papers from the Dean of Faculty
- Maintain an attendance register which shall be completed and lodged with the Dean of Faculty at the end of the examination
- Receive from invigilators, report on any misconduct observed or suspected and in turn report the same to the Dean of Faculty after the examination
- Deposit answer scripts with the Head of Department immediately after the examination
- Ensure that:
  - Within the first 30 minutes of the examination, no candidate leaves the examination room except on cases of emergency
  - After the first 60 minutes of the examination, no candidate enters the examination room
  - During the last 15 minutes of the examination, no candidate leaves the examination room.

H. Administration of Examination

Conduct

a) Students shall be at the examination venue at least 30 minutes before the advertised time for the examination. Students MUST BE PROPERLY DRESSED before entering the examination room.
b) Students must produce their identity card on entry to every examination and display them conspicuously throughout the duration of the examination. It shall be the duty of the invigilator (s) to ensure that students write their names, registration numbers and other required information in the attendance register
c) Students shall complete all information required of them in the answer booklets including continuation sheets
d) The scripts of students who leave the examination room during emergency must be retrieved by the Invigilator before the student leaves the room.
e) The Invigilator shall arrange the answer scripts in labeled envelopes and submit the same to the Chief examiner or course coordinator 30 minutes after the completion of the examination. It shall be the responsibility of the Chief Examiner or Course Coordinator to collect and confirm the number of scripts received from the Chief invigilator.

Misconduct

- Obstructing any invigilator or any other examination official in the performance of his/her duties, thereby causing harm or damage
- Impersonation (writing examination for another)
- Destroying evidence/exhibits by chewing or by any other means or an attempt to do same
- Use of scripts other than those designated for an examination
- Use of answer booklet consisting mainly of loose or continuous sheets
- Bringing into the examination hall any unauthorized materials/gadgets relevant to the examination being taken
- Copying with or without cooperation
- Presentation of identical responses to the same question by two or more students even in cancellation
- All forms of communication during examinations (passing papers or dropping papers, conscious or unconscious)
- Appearance of different handwritings in a student's scripts
- Courier (smuggling of examination question papers out of the examination room
- Reading of notes/textbooks in the in convenience during the examination or under any guise
- Falsification of test or examination marks/grade
- Failing to sign in examination misconduct form when caught cheating in an examination
- Plagiarism: Copying and downloading other people's work for the project or assignment

Invigilators shall ensure that affected students complete the examination misconduct form for all established cases of misconduct. Such students shall be allowed to continue with the examination after documenting the misconduct accordingly.

L. Procedure for Handling Examination Misconduct

- The student is made to state and sign the examination misconduct form with his/her own version of the case. Thereafter, the student is allowed to continue to write that examination using another answer booklet. The form should be counter-signed by a student witness if available. In the case the student refuses to sign the misconduct form, the security unit will be invited to remove the said student from the examination hall.
- The alleged offender's answer booklet will be retrieved by the Chief Invigilator and stapled to the signed form for onward transmission through that student's department to the Examination Misconduct Committee. The latter will then decide the level of the misconduct committed and the corresponding penalty.
- The penalty ranges from cancelling of the student's paper to rustication for one year.
- The Chief Invigilator should write a report packaging the defaulter's booklet to be forwarded to the Examination Misconduct Committee for appropriate action.

J. Appointment and Duties of the Examination Officer

i. Appointment

Each department shall appoint an Examination Officer not below the rank of Lecturer 1. The appointment is for two academic sessions, subject to reappointment for another period of two years. The Head of Department, through the Head of the Department applies to Senate for approval of the appointment after which the Registrar issues a letter of appointment.

ii. Duties

It is the duty of the Examination Officer to:

- Prepare the timetable, arranging for invigilation, and other requirements necessary for conducting all examinations in the Department
- Prepare a summary result sheet for all students registered in the department

K. Duties of Examiners

1. Internal Examiners

Internal examiners are Lecturers who taught a course and set the examination questions on the course. Their duties include:

- Setting and participation in the moderation of examination questions in the department
- Invigilating all examinations of the courses they teach
- Grading all examination scripts of the courses they taught and submitting the grades to the Head of Department
- Taking part in the meeting of the Department to consider and approve results for submission to the Department Board

2. External Examiner

An External Examiner is a senior academic appointed by the University who is currently not serving in the University. Usually those appointed are proven academics with a known track record in an appropriate area of specialization in the department for a period as may be determined by Senate. The duties of the External Examiner include:

- Moderation and vetting of the course outlines for all final year courses
- Moderation and vetting of all the examination questions as well as answer scripts of all final year students
- Moderation and vetting of final research projects/long essays
- Resolving any conflict in grades for final year students
- Vetting of final year grades and final degree of pass
- Writing a report covering aspects of his/her moderation/vetting

L. Appeals for Re-assessment of Examination Scripts

- Students may appeal to the Registrar through their Heads of Department for reassessment of their examination scripts on payment of a fee as stipulated by Senate
- Appeals for the reassessment of scripts can only be made by the student(s) concerned
- Group appeals by all candidates involved in a particular examination cannot be entertained
- If the appeal results in significant improvement (change in letter grade) on the student's original grade, the appeal fee shall be refunded to the appellant.

For an appeal to be valid, notice in writing by the student to the Head of Department must be lodged with the Registrar within four weeks of the publication of the relevant results. The re-assessor must not be known to the aggrieved student. The re-assessor shall be furnished with the following:

- Two scripts whose scores are around the petitioner's score
- The petitioner's script
- The marking scheme used to assess the script(s) by the course lecturer.

There should be no indication to the re-assessor which script(s) require(s) special attention. After the exercise, only the petitioner's scrip(s) will be affected by any change in grade while the other script(s) retain(s) their original grades.