DEPARTMENT HANDBOOK

Bachelor of Science [B.Sc.]

in

SOFTWARE ENGINEERING

[2022/2023 SESSION]

AFRICAN UNIVERSITY OF SCIENCE AND TECHNOLOGY

CONTENTS

1.0 INTRODUCTION TO AFRICAN UNIVERSITY OF SCIENCE AND TECHNOLOGY (AUST)

Although situated in Nigeria, The African University of Science and Technology, Abuja (AUST-Abuja) was set up to be a pan-African university to serve all of sub-Saharan Africa. It is the first of a small group of institutions created to be part of a framework called the African Institute of Science and Technology (AIST), others are now to be found in Burkina Faso and Arusha, Tanzania. The AIST concept drew its inspiration from the highly successful Indian Institutes of Technology (IIT) and the Indian Institute of Science (IIS). AUST is a respected, not-for-profit world-class technological university, whose purpose is to train, and help retain in Africa, top-level scientists and engineers. The medium of instruction at AUST-Abuja is English. Special language courses are provided for non-English speakers to prepare them for participation in programs in English. The AUST-Abuja campus is located within the Abuja Technology Village (ATV), a planned cluster of leading research institutions and technology companies, located on a 1,000-hectare site where it occupies 240 hectares. The land was donated by the Government of Nigeria for this purpose. It is conveniently situated just 10 minutes from Abuja's city centre and 20 minutes from Nnamdi Azikiwe International Airport. AUST received accreditation from the Nigerian Universities Commission (NUC) in 2007 and received its first batch of students in 2008. Since then, true to the philosophy behind its establishment, as a pan-African institution to build science, engineering and technology capacity in all of sub-Saharan Africa, AUST has educated students from more than 26 African countries in East, West, Central and Southern Africa

In its short 16 years of teaching, research and innovation, AUST has already matured into a promising institution with a combination of Resident Faculty, over 40 full-time staff members, 80 visiting Faculty, world renowned specialists in their fields, brought in every year to teach and supervise research for over 160 MSc/PhD students in Computer Science, Materials Science and Engineering, Petroleum Engineering, Pure and Applied mathematics and Theoretical and Applied Physics. It is not surprising that AUST is already emerging as a noteworthy research university in the heart of Africa; the University has already become one of the very few African universities designated as a Center of Excellence by the World Bank, with a special focus on its programs in Material Science and Engineering and thus serves as the coordinating center for the region's Pan African Materials Institute (PAMI). Our groups in materials science and physics are developing the next generation of organic solar cells and light emitting devices. They are also developing the next generation of nanoparticles now being used in the detection and treatment of such medical conditions as cancer and cardiovascular disease. Similarly, our groups in petroleum engineering are working on using cutting-edge theoretical and computational tools to develop new tools for the oil and gas industry. These include theoretical and computational tools for flow assurance and reservoir management, as well as the introduction of nano-mechanics and fracture mechanics to the management of pipelines, reservoirs and offshore structures.

In the area of computer science, our DEVS team has won a number of top awards at International Conferences and professional events. The groups in computer science are also working on e-learning platforms and wireless technologies that are being integrated with mobile telecommunications. This is being done in an environment that has a 64-node multi-processor that has been used as the basis for the setting up of a high-performance computing (HPC) array. Our mathematics group has established

itself as one of the leading groups in functional analysis. With its recent focus on fixed point theory and equations of the Hammerstein type, they are solving problems at the frontiers of mathematics.

The fundamental objective of the African University of Science and Technology (AUST) is to educate the next generation of African scientists and engineers – providing them with the technical and leadership capacities to solve real African problems and contribute to the economic and social transformation of the continent. AUST will deliver a strong foundation in the fundamentals of mathematics and science; a deep understanding of the research frontiers; and an orientation towards entrepreneurship and service. AUST is committed to excellence in teaching, research and service. It is also motivated to build collaborative partnerships with for-profit and not-for-profit institutions to support the millennium goals for the development of the African Continent. The goal of the curriculum is to prepare students to become very high-level professionals who can play a transformational role in African industry/business and academia.

2.0 INTRODUCTION TO THE SOFTWARE ENGINEERING PROGRAMME

The discipline of Software Engineering (SE) focuses on producing graduates who are ready to develop and maintain quality software systems of scale, for organizations and businesses, within the constraint of time, budget and other requirements. In addition to its core computer science technical foundation, SE also involves human processes that are harder to formalize than the logical abstraction of computer science. The discipline therefore borrows and adapts from traditional engineering practice as well as from the field of project management. A focus of a SE specific curriculum must be able to develop students into software engineers through course work, practical and applied project experience.

2.1 Mission

The mission of the program is impacting the training of software engineering to produce graduate suitable for job market, post-graduate work and capable of applying their knowledge and skill to solve complex social and technological problems across the globe.

2.2 Vision

To produce competent graduates that will compete with their counterpart in the development of Africa and the country at large.

2.3 Philosophy

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 context 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 program in SE is aimed at achieving the goals and objectives of the National policy on Industrialization 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 SE.
ii. Practical exposure to application of SE to problem solving.
iii. Adequate training in human and organizational behavior and management in the software development cycle.
iv. Developing in the product entrepreneurial knowledge, a sense of public responsibility and spirit of self-reliance.
v. Nurturing of partnership between the institution and the SE for effective program delivery.
vi. Creating an awareness and understanding of the moral, ethical, legal and professional obligations needed to function as part of the computing ecosystem while protecting human health and welfare and environment in a global society.
vii. Creating an awareness and understanding of the need to develop leadership and team building skills to maximize the benefits of SE education and its application to solving problems.

2.4 Aim and Objectives

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

a. To appreciate the importance of Computer Science (as base) in such areas as principals of programming, algorithm, data strictures, database and programming language.
b. To develop and utilize the practical skills acquired in software architecture and design, software metrics, V&V, requirements and analysis, and the software engineering process for the production of software-based systems.
c. To develop expertise in programming languages with the emphasis on production of robust, reliable and cost-effective software.
d. 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 analyst, developers, consultant etc.
e. To be able to produce and manage high quality software-based solutions with long life cycles especially for large complex systems.
f. To develop the understanding and engineering skills needed to become the architects and project leaders building systems in which software plays a critical role.
g. To leverage on SE as driving force behind the new technologies that are transforming the way we live and work.
h. To have the requisite knowledge and skill base for further academic and professional development in SE
i. The programme in SE 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.

2.5 Learning Outcome

By the time of their graduation, all students in the software engineering program will have demonstrated the ability to:

a. Apply discrete mathematics, statistics, physical science, computer science and related disciplines to model and analyze proposed and existing software systems.
b. Apply quality principles to the definition and evaluation of software systems and processes in order to verify and validate software systems.
c. Analyze and design, implement and maintain complex software systems using contemporary analysis and design principles such as cohesion and coupling, abstraction and encapsulation, design patterns, frameworks and architectural styles.
d. Apply principles of team process and project management to develop a software system. This includes the ability to assume distinct operational roles (e.g., configuration management, quality assurance) in addition to design and implementation.

e. Demonstrate knowledge of economics, humanities and social sciences that are related to existing software systems, as well as potential consequences of proposed systems with a clear understanding of the ethical and professional responsibility incumbent upon them in different software development organizations (e.g., commercial off-the-shelf vs. safety-critical systems).
f. Be able to analyze, document and track system requirements. This includes writing concise and correct documents relating to all phases of the software development cycle and make oral presentations of technical material.
g. Explore new topics in software engineering or related application domains with limited oversight and input from faculty or mentors.
h. Rapidly and independently learn, assess, and adapt to new languages, environments, and paradigms for software development.
i. Be able to work in one or more application domains and to relate principles of software engineering to at least one domain where those principles are or can be applied.
j. Have an awareness of current industry standards and practices.
k. Have strong oral and written communication skills.
l. Understand professional responsibility and the application of ethical principles.

2.6. Attainment Levels

Graduates of Software Engineering are expected to have the ability to apply knowledge and skills to solving theoretical and practical problems in Software Engineering, development of relevant ICT for national development and societal needs.

2.7 Admission Requirements

2.7.1 Entry Requirement and Duration

The entry requirements are at least credit level passes in five subjects to include English Language, Mathematics/Further Mathematics, and any of Physics, Data Processing, Computer Science, plus two (2) other science subjects, at the Senior Secondary School Certificate or its equivalent.

2.7.1.1 Nigerian Students

The entry requirements shall be at least credit level passes in five subjects to include English Language, Mathematics/Further Mathematics, and any of Physics, Data Processing, Computer Science, plus two (2) other science subjects, at the Senior Secondary School Certificate or its equivalent. In addition, an acceptable pass in the Unified Tertiary Matriculation Examination (UTME) is required for admission into 100-level. Candidates are admitted into the degree programme in any of the following three ways:

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

2.7.1.2 UTME Entry Mode

The minimum academic requirement is credit level passes in five subjects at O'Level in national Examination/Cambridge O'Level: Mathematics/ Further Mathematics, English Language and three (3) other Science subjects.

UTME SUBJECTS: Physics, Mathematics and Chemistry/biology

2.7.1.3 Direct Entry Mode

- In addition to O'Level requirements stipulated above, applicants should possess at least two A'Level papers in Level in relevant subjects (Chemistry, Mathematics and Physics, Biology) may be considered for admission into 200-level.
- OND in relevant discipline with at least upper credit grade in addition to the five credit passes as stated above.
- HND in relevant discipline with at least upper credit in addition to five credit passes as stated above.
- Joint Universities Preliminary Examinations Board (JUPEB)/ Interim Joint Matriculation Board Examination (IJMBE) and AUST Foundation Program are also acceptable.

2.7.1.4 Foreign Students

- The University proposes to admit foreign students through the options listed below and will comply with the admission policy into Nigerian Universities and JAMB.
- Scholastic Aptitude Test (SAT) – An international entrance exam for admission into Universities in the United States and American University abroad, which is conducted in over 130 countries as
- Any other internationally recognized University Entrance Exam, equivalent of the JAMB in Nigeria.
- Joint Universities Preliminary Examinations Board (JUPEB)/ Interim Joint Matriculation Board Examination (IJMBE) and AUST Foundation Program are also acceptable.

2.7.2 Duration

A student will not be allowed to exceed an additional 50 per cent of the duration of the programme if he fails to graduate within the minimum number of years.

2.7.2.1 UTME

Four (4) academic sessions or eight (8) semesters.

2.7.2.2 Direct Entry

Three (3) academic sessions or six (6) semesters. In general, no student will be allowed to exceed an additional 50% of the normal duration of the programme.

2.8 Graduation Requirements

To qualify for the 4-year Bachelor of Science (B.Sc.) degree award in the Software Engineering programme, a student must pass the minimum number of credit units as shown in the Tables 1-3. This table is based on Minimum Academic Standards and guidelines of National Universities Commission (NUC).

2.8.1 Course Credit System

Students in department of computer science are expected to take a minimum of 160 credit units for the award of a Bachelor's degree in Software Engineering. A minimum of 15 credit units and a maximum of

twenty-four (24) credit units should be taken by each student per semester. Each of the courses in the programme is expected to be taught for a semester which will last for a minimum of fifteen (15) weeks. The teaching should be distributed into lectures, tutorials and workshops/studio practical.

2.8.2 Grading of Courses

The grading system provided by the University is an indicative policy for the calculation of grade point average (GPA). The grading system policy for students at AUST showing the letter grades, its corresponding grade point and the score range is shown below:

Table 1: Grading System at AUST

Grades on 4.0 Scale	Letter Grade	Score Out of 100
4.00	A	95 – 100
3.75	A-	89 – 94
3.25	B+	83 – 88
3.00	B	77 – 82
2.75	B-	71 – 76
2.25	C+	65 – 70
2.00	C	59 – 64
1.75	C-	53 – 58
1.00	D	48 – 52
0	F	0 – 47

2.8.3 Grade Point Average and Cumulative Grade Point Average

For the purpose of determining a student's standing at the end of every semester, the Grade Point Average (GPA) system shall be used. The GPA is computed by dividing the total number of Units x Grade Point (TUGP) by the total number of units (TNU) for all the courses taken in the semester. The Cumulative Grade Point Average (CGPA) over a period of semesters is calculated in the same manner as the GPA by using the grade points of all the courses taken during the period. Calculation of GPA or CGPA is shown in Table 2.

Table 2: GPA Calculation

Course	Units	Grade Point	Units x Grade Point (UGP)
C_1	U_1	GP_1	$U_1 \times GP_1$
C_2	U_2	GP_2	$U_2 \times GP_2$
-	-	-	-
-	-	-	-
C_i	U_i	GP_i	$U_i \times GP_i$
-	-	-	-
-	-	-	-

C_N	U_N	GP_N	U_N x GP_N
TOTAL	TNU		TUGP

$$TNU = \sum_{i=1}^{N} U_i \qquad TUGP = \sum_{i=1}^{N} U_i * GP_i \qquad CGPA = \frac{TUGP}{TNU}$$

2.8.4 Degree Classifications

Classes of degree are to be awarded depending on the cumulative GPA obtained. The classes of degrees that may be awarded are First Class Honours, Second Class Honours (Upper Division), Second Class Honours (Lower Division) and Third Class Honours. Table 3 shows the degree classification.

Table 3: Degree Classification

CGPA	CLASS OF DEGREE
3.50 – 4.00	First Class Honours
2.50 – 3.49	Second Class Honours (Upper Division)
1.40 – 2.49	Second Class Honours (Lower Division)
0.50 – 1.39	Third Class Honours

2.8.5 Probation

Probation is a status granted to a student whose academic performance fall below an acceptable standard. A student whose Cumulative Grade Point Average is below 0.50 at the end of a particular year of study, earns a period of probation for one academic session.

2.8.6 Withdrawal

- A student shall be requested to withdraw from a programme if at the end of a probation period, the student still does not make satisfactory progress. Such student shall be at liberty to apply for a change of programme within the University.
- Subject to the conditions for withdrawal and probation, a student may be allowed to repeat the failed course Unit(s) at the next available opportunity, provided that the total number of credit units carried during that semester does not exceed 24, and the Grade Points earned at all attempts shall count towards the CGPA.

2.9 Evaluation

2.9.1 Techniques of Students Assessment

The evidence, on which the assessment of a student's achievement is based, will include the following:

- Formal examinations
- Laboratory Reports
- Problem Solving Exercises
- Oral Presentations
- Essay Assignments/Term Papers

- Collaborative Project Work
- Individual Project Work
- Report on External Placement (SIWES)
- External Examiners Report
- Surveys and Evaluations

2.9.2 External Examiner's System

External examiners shall be appointed once in a year particularly at the end of each session to moderate examination questions, review the scripts of the students, and provide an overview of the work of the students in all classes, particularly those in the final year. It shall be mandatory on the Chief Examiner to review the questions set by his colleagues before those questions are forwarded to the External Examiner in order to ensure that they reflect the coverage of the syllabi and the manner in which they were taught. The system also provides avenues for assessing comparability of programmes and the maintenance of minimum standards.

2.9.3 SIWES Rating and Assessment

The Nigerian Universities Commission (NUC) has mandated and approved Students Work Experience Programme (SWEP) and Students Industrial Work Experience Scheme (SIWES), for Nigerian Universities and other institutions of higher learning; for its students to undergo various training in their respective fields of studies due to lack of relevant facilities and machinery within the Nigerian Universities. Thus, mandatory for all students offering relevant courses in higher institutions to undergo the SWEP and SIWES programme at their assigned level and stipulated time. The exposure to a combination of field and office experience both in the public and private sectors and/or construction activities relevant to their individual disciplines. All students in the Chemistry discipline will be exposed to a period of compulsory, supervised SIWES in addition to Laboratories/Workshop/Practical/Studio Training as reflected in the individual programmes. Such training shall be undertaken in an approved establishment. A minimum period of a semester is considered to be adequate. The student is expected to submit a systematic log-book for assessment at the end of the training period. Students with unsatisfactory performance shall be required to repeat the training programme.

2.9.4 Students' Evaluation of Courses

At the end of every semester, students shall be given the opportunity to evaluate the courses taken in the semester based on the following criteria:

- relevance
- adequacy in terms of time and content coverage
- students understanding of the courses
- adequacy of lectures, tutorials and practical
- standards of continuous assessment and examinations

2.9.5 Maintenance of Curricula Relevance

The various curricula for the Pure and Industrial Chemistry should be reviewed from time to time as reflected in each individual programme. General review will be conducted every five (5) years, in full consultation with the relevant professional bodies.

2.9.6 Performance Evaluation Criteria

The general performance indices useful to accreditation assessors and for internal review terms are as specified in each individual programme, especially as these relate to the following: staff/student ratio, facilities such as laboratories, workshops, library/Information and Communication Technology (ICT), staff composition and minimum space requirements.

3.0 RESOURCES

3.1 Staffing

	ACADEMIC STAFF				SNR. TECH. STAFF	SNR. ADMIN. STAFF		JUNIOR STAFF	
	PROF.	READER/ ASSOC. PROF.	SNR. LECT.	LECT. 1 & BELOW		SEC.	NON-SEC.	TECH.	NON-TECH.
Core Staff on the ground for the programme	2	2		6	3	1	1	1	1
Staff available for the programme from other source (s)	2	4	2						
Total	4	6	2	6	3	1	1	1	1

Table 4: List of Existing Academic Staff for The Programme

S/N	NAME OF ACADEMIC STAFF	AREA OF SPECIALIZATION	DISCIPLINE	QUALIFICATION	RANK
1	Rajesh Prasad	Operating system; Software Engineering	Computer Science	PhD	Associate Professor (FT) and HoD
2	Kehinde Grace Samuel	Data Modelling	Computer Science	PhD	Senior Lecturer (FT)- Program Coordinator
3	Mohamed Hamada	Compiler	Computer Science	PhD	Professor (PT)
4	Amit Mishra	Software Engineering, Systems Analysis	Computer Science	PhD	Senior Lecturer (PT)
5	Hauwa Amshi	Artificial Intelligence	Computer Science	MSc	Lecturer I(FT)
6	Toluwase Sunday	Artificial Intelligence	Computer Science	MSc	Lecturer I(FT)
7	Mubaraka Sani	Artificial Intelligence	Computer Science	MSc	Lecturer I(FT)

8	Yoro Diouf	Modeling and simulation	Computer Science	MSc	Lecturer I (FT)
9	Hajara Innyah Abdulwahab	Modeling and simulation	Computer Science	MSc	Lecturer I (FT)
10	Effiong Blessing Uduakobong	Machine Learning	Computer science	MSc	Assistant lecturer (FT)
11	Umoru Ibrahim	Data mining	Computer science	MSc	Assistant lecturer (FT)
12	Ugochi Orji	Software Engineering	Computer science	MSc	Assistant lecturer (FT)

FT – Full time PT – Part time

Table 5: List of Existing Non-Academic Staff for the Programme

S/N	NAME OF NON-ACADEMIC STAFF	AREA OF SPECIALIZATION	DISCIPLINE	QUALIFICATION	RANK
1	Abiodun Bidemi	Networking	ICT	BSc	System administrator
2	God-dey Chukwuemeka Onyele	Hardware and Troubleshooting	ICT	BSc	IT-Assistant
3	Kasiemobi Martins Offie	Programmer	ICT	BSc	Programmer
4	Achimugu Amichi Nafisat			BSc	Secretary

3.2 Library Facility

The library is the heart of teaching and research. The central library has a huge collection of books and bound periodicals. Department is also having a departmental library. In order to facilitate all the readers in selecting the reading materials of their choice, the access to stacks is open to its members.

It works as nerve center of the institution by keeping the knowledge of students and faculty members updated. Information data bank is constantly updated and facilities are added. The central library is equipped with Ebscohost database, which contains large number of e-books and e-journals. Some e-journals are: ACM Transactions and IEEE Transactions etc. E-books are available for most of the international publishers.

3.3 ICT

The responsibilities of the Information and Communication Technology (ICT) unit comprise: the development and maintenance of the AUST ICT infrastructure; the provision of ICT advice for projects; the management of online services and databases.

3.4 Laboratory

AUST has two dedicated computer lab (20 computers each), one advanced lab (10 high configuration Apple computers) and one High performance computer (HPC), which provides computer services to the

students. Computer labs are typically provided by libraries to the public, by academic institutions to students who attend the institution, or by other institutions to the public or to people affiliated with that institution.

4.0 COURSE CONTENT/SYLLABUS

4.1 Course Structure Software Engineering

Course Structure at 100 Level

Course Code	Course Title	Units	Status	LH	PH
CHM 101	General Chemistry	3	R	45	-
SEN 101	Introduction to Computer Science	3	C	30	45
SEN 102	Introduction to Problem Solving	3	C	30	45
SEN 104	Computer Programming I	3	C	30	45
GST 111	Communication in English I	2	C	30	-
GST 112	Logic, Philosophy & Human Existence	2	E	30	-
GST 113	Nigerian Peoples and Culture	2	R	30	-
GST 121	Use of Library, Study Skills and ICT	2	R	30	-
MTH 101	General Mathematics I	3	R	45	-
MTH 102	General Mathematics II	3	R	45	-
MTH 103	General Mathematics III	3	R	45	-
PHY 101	General Physics I	3	R	45	-
PHY 102	General Physics II	3	R	45	-
PHY 107	General Physics Practical I	1	R	-	45
PHY 108	General Physics Practical II	1	R	-	45
CHM 102	Chemistry Practical	1			
	TOTAL	38			

Course Structure at 200 Level

Course Code	Course Title	L	T	P	Units
SEN 201	Introduction to Software Engineering	3	0	0	3
SEN 202	Computer Programming II	3	0	0	3
SEN 203	Discrete Structures and Theory of Logics	3	0	0	3
SEN 205	Software Requirements and Design	3	0	0	3
SEN 206	Software Construction	3	0	0	3
SEN 207	Computer Organization and Architecture	2	1	0	3
SEN 208	Object oriented analysis and design	3	0	0	3
SEN 209	Data Structures and Algorithms	2	0	1	3
SEN 210	Operating Systems	3	0	0	3
SEN 211	Software Engineering Process	3	0	0	3
SEN 212	Computer Security	3	0	0	3
GST 211	Environment and Sustainable Development	2	1	0	3
GST 223	Introduction to Entrepreneurship	2	0	9	2

STA 201	Statistics	3	0	0	3
	TOTAL				39

Course Structure at 300 Level: Software Engineering

Course Code	**Course Title**	L	T	P	Units
SEN 301	Design and Analysis of Algorithms	3	0	0	3
SEN 303	Software Testing and Quality Assurance	3	0	0	3
SEN 305	Web Application Development	2	0	1	3
SEN 307	Database Systems	2	0	1	3
SEN 309	Concepts of Programming Languages	2	0	1	3
SEN 311	Machine Learning Techniques	3	0	0	3
GST 324	Leadership skills	2	0	0	2
SEN 399	SIWES				6
	TOTAL				26

Course Structure at 400 Level: Software Engineering

Course Code	**Course Title**	L	T	P	Units
SEN 401	Software Configure Management and Maintenance	3	0	0	3
SEN 402	Software Engineering Economics	3	0	0	3
SEN 403	Software Engineering Project Management	3	0	0	3
SEN 404	Human Computer Interaction	3	0	0	3
SEN 405	Research Methodology	2	0	0	2
SEN 406	Open-Source Software Development and Applications	3	0	0	3
SEN 407	Software Engineering Professional Practice	1	1	0	3
SEN 408	Distributed, Parallel and Cloud Computing	2	0	1	3
SEN 409	Software Engineering Security	2	0	1	3
SEN 410	Software Architecture and Design	2	1	0	3
SEN 499	Final Years Project			18	6
	TOTAL	27			35

Electives: 6 Units to be selected from

Course Code	**Course Title**	L	T	P	Units
SEN 411	Artificial Intelligence	2	0	1	3
SEN 412	Special Topics in Software Engineering	3	0	0	3
SEN 413	Engineering Mobile Applications	3	0	0	3
SEN 414	Fault Tolerant Computing	1	0	1	2

SEN 415	Game design and Development	2	0	1	3
SEN 416	Modeling and Computer Simulation	3	0	0	3

4.2 Course Synopses

SEN 101: Introduction to Computer Science

Survey of computers and information processing and their roles in society. This course introduces a historical perspective of computing, hardware, software, information systems, and human resources and explores their integration and application in business and other segments of society. Students will be required to complete lab assignments using the PC's operating system, and several commonly used applications, such as word processors, spreadsheets, presentations, graphics and other applications. Internet and on-line resources, browsers and search engines.

SEN 102: Introduction to Problem Solving

Role of Algorithms in problem solving process, concepts and properties of Algorithms. Implementation strategies, Development of Flow Charts, Pseudo Codes. Program objects. Implementation of Algorithms in a programming Language – Visual BASIC/JAVA/C/C++

SEN 104: Computer Programming I

Introduction to problem solving methods and algorithm development, designing, coding, debugging and documenting programmes using techniques of a good programming language style, programming language and programming algorithm development. A widely used programming language should be used in teaching the above.

SEN 201: Introduction to Software Engineering

Principles of software engineering: Requirements, design and testing. Review of principles of object orientation. Object oriented analysis using UML. Frameworks and APIs. Introduction to the client-server architecture. Analysis, design and programming of simple clients and servers. Introduction to user interface technology.

SEN 202: Principles of Programming II

The course is a continuation of the SEN 2301 course with intermediate programming principles and mechanisms emphasizing advanced object-oriented techniques including inheritance, polymorphism, and interfaces; exception handling, design patterns, simple GUI programming, multi-threaded programming, abstract & dynamic containers such as linked lists, stacks, queues, and trees and their associated algorithms including those based on recursion.

SEN 203: Discrete Structures and Theory of Logics

Basic Set Theory: Basic definitions, Relations, Equivalence Relations Partition, Ordered Sets. Boolean Algebra & Lattices, Logic, Graph theory: Directed and Undirected graphs, Graph Isomorphism, Basic Graph Theorems, Matrices; Integer and Real matrices, Boolean Matrices, Matrices med m, Path

matrices. Adjacency Vectors/Matrices: Path adjacency matrix, Numerical & Boolean Adjacency matrices. Applications to counting, Discrete Probability Generating Functions.

SEN 205: Software Requirements and Design

Definition of a software requirement, product and process requirements, functional and non-functional requirements, emergent properties and quantifiable requirements. Languages and models for representing requirements. Analysis and validation techniques, including need, goal, and use case analysis. Requirements in the context of system engineering. Specifying and measuring external qualities: performance, reliability, availability, safety, security, etc. Specifying and analyzing requirements for various types of systems: embedded systems, consumer systems, web-based systems, business systems, systems for scientists and other engineers. Resolving feature interactions. Requirements documentation standards. Traceability. Human factors. Requirements in the context of agile processes. Requirements management: Handling requirements changes.

SEN 206: Software Construction

This course will be an in-depth study of software design and implementation using a modern, object-oriented language with support for graphical user interfaces and complex data structures. Special emphasis will be placed on critical issues for consideration in modern 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. Topics will include specifications, abstraction techniques including typing, access control, inheritance, polymorphism, genericity and design patterns, frameworks & 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.

SEN 207: Computer Organization and Architecture

Memory system, general; characteristics of memory operation. (Technology-magnetic recording semi-conductor memory, coupled devices, magnetic bubble). Memory addressing, memory hierarchy, virtual memory control systems. Hardware control, micro programmed control, Asynchronous control, i/c control. Introduction to the methodology of faulty tolerant computing.

SEN 208: Object Oriented Analysis and Design

An introduction to object-oriented software engineering (OOSE) as a better alternative to traditional methods of software engineering is presented. It reviews important issues of software engineering especially that of managing complexity discusses how the object approach makes a better alternative. It includes the basic concepts, principles and diagramming notations of object-oriented analysis and design (OOAD). The course discusses OO tools such as OO programming languages, OO CASE tools, OO database management systems, and tools for future reliable software. It looks at the implications for the future of the software industry and will include working knowledge and experience of at least one OO CASE tool or OO DBMS.

SEN 209: Data Structures and Algorithms

The course covers, from an object-oriented programming language perspective, fundamental data structures, algorithms for manipulating and retrieving information from these data structures, and techniques for analyzing their efficiency in terms of space and time. The distinction between an Abstract Data Type and its implementation is emphasized. Topics include recursion, complexity analysis, linear data structures (stacks, queues, priority queues, lists and strings), and 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.

SEN 210: Operating Systems

This course 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.

SEN 211: Software Engineering Process

This course will expose students to various software engineering processes models, in particular, Waterfall, RUP, MBASE, Prototyping paradigms, V-process model, peer-programming, etc.

SEN 212 Computer Security

Information management, Computer systems, Risk management, Cyber law and ethics, technical report writing, Investigation techniques, encryption and decryption

SEN 301: Design and Analysis of Algorithms

Basic algorithmic analysis: Asymptotic analysis of Upper and average complexity bounds; standard Complexity Classes Time and space tradeoffs in algorithms analysis recursive algorithms. Algorithmic Strategies: Fundamental computing algorithms: Numerical algorithms, sequential and binary search algorithms; sorting algorithms, Binary Search tress, Hash tables, graphs & its representation.

SEN 303: Software Testing and Quality Assurance

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.

SEN 305: Web Application Development

Covers client server model for web applications and associated client-side and server-side technologies, MVC development guidance and development of a complete web application using a framework such as Ruby of rails of Django.

SEN 307: Database Systems

Information storage & retrieval, Information management applications, Information capture and representation, analysis & indexing, search, retrieval, information privacy; integrity, security; scalability, efficiency and effectiveness. Introduction to database systems: Components of database systems DBMS functions, Database architecture and data independence use of database query language. Rational Databases: Mapping conceptual schema to relational Schema; Database Query Languages (SQL) Concept of Functional dependencies & multi-valued dependencies, transaction processing; distributed databases.

SEN 309: Concepts of Programming Languages

Overview of programming languages: History of programming languages, Brief survey of programming paradigms (Procedural languages, Object-oriented languages, Functional languages, Declarative – non-algorithmic languages, Scripting languages), the effects of scale on programming methodology; Language Description: Syntactic Structure (Expression notations, abstract Syntax Tree, Lexical Syntax, Grammars for Expressions, Variants of Grammars), Language Semantics (Informal semantics, Overview of formal semantics, Denotation semantics, Axiomatic semantics, Operational semantics); Declarations and types: The concept of types, Declaration models (binding, visibility, scope, and lifetime), Overview of type-checking, Garbage collection; Abstraction mechanisms: Procedures, function, and iterations as abstraction mechanisms, Parameterization mechanisms (reference vs. value), Activation records and storage management, Type parameters and parameterized types, Modules in programming languages; Object oriented language paradigm; Functional and logic language paradigms.

SEN 311 Machine Learning Techniques

Introduction to Machine Learning, Supervised Learning and Linear Regression, Classification and Logistic Regression, Decision Tree and Random Forest, Naïve Bayes and Support Vector Machine, Unsupervised Learning.

SEN 401: Software Configure Management and Maintenance

Management of the software configuration management (SCM) process, organization context, constraint and guidance, plan, and surveillance. Software configuration management identification and software library, software configuration control, requesting, evaluating and approving software changes, implementing software changes and deviation of v=waivers, software configuration accounting, SCM status info and reporting, SCM auditing. Key issues in software maintenance, technical issues, management issues, maintenance cost estimation, and software maintenance measurement; maintenance processes and activities; Techniques for maintenance-program comprehension, reengineering, reverse engineering, migration and retirement.

SEN 402: Software Engineering Economics

The scope of engineering economics; microeconomics; supply, demand, and production; cost benefit analysis and break-even analysis; return on investment; analysis of options; time value of money; management of money: economic analysis, accounting for risk.

SEN 403: Software Engineering Project Management

This course will give students an understanding of the most common processes, tools, techniques, and theories that are necessary to manage IT projects especially those regarding software. Managing IT projects poses some additional unique challenges due to several factors including: (1) the rapid pace of technological changes occurring in the IT field, (2) the invisible nature of software, (3) the ever-present pressure to add new features and functionality to systems, and (4) the difficulty of managing the organizational changes that accompany most IT implementations. The course examines the defining characteristics of IT projects and introduces the student to a variety of project management techniques that must be applied in a disciplined approach to IT project management. Managing IT projects that follow both plan-driven traditional development methods as well as agile methods will be covered. Key topics include: Project management tools such as MS Project; analysis of options and risks; project planning; cost estimation & productivity metrics; scheduling; factors influencing productivity and success; release and configuration management; management of expectations; planning for change; software process standards; process implementation; software contracts and intellectual property; approaches to maintenance & quality assurance; project reporting and case studies of real industrial projects.

SEN 404: Human Computer Interaction

Introduction of HCI, Focusing on tree key areas: Design, Implementation, and evaluation.

Design: How to design good interface, starting with human capabilities and using those capabilities to drive design techniques: task analysis, user centered design, iterative design, usability guidance, interaction styles and graphic design principles.

- Implementation: Technique for building user interfaces, including low fidelity prototypes, Wizard of Oz, and other prototyping tools; input models, model view controller, layout, constraints, and toolkits.
- Evaluation: Techniques for evaluating and measuring interface usability, including heuristic evaluation, predictive evaluation, and user testing.

SEN 405: Research Methodology

Introduction to research – The role of research, research process overview Philosophies and the language of research theory building – Science and its functions, What is theory?, and The meaning of methodology Thinking like a researcher – Understanding Concepts, Constructs, Variables, and Definitions Problems and Hypotheses – Defining the research problem, Formulation of the research hypotheses, The importance of problems and hypotheses Research design – Experimental and Non experimental research design, Field research, and Survey research, Methods of data collection – Secondary data collection methods, qualitative methods of data collection, and Survey methods of data collection, Attitude measurement and scaling – Types of measurement scales; Questionnaire designing – Reliability and Validity, Sampling techniques – The nature of sampling, Probability sampling design, Nonprobability sampling design, Determination of sample size, Processing and analysis of data, Ethical issues in conducting research, Report generation, report writing, and APA format – Title page, Abstract, Introduction, Methodology, Results, Discussion, References, and Appendices

SEN 406: Open-Source Software Development and Applications

Introduction, principles and application, software development processes. Covers economy, business, and societal and intellectual property aspects of open-source software. Hands on experiences on open-source software and related tools through developing various open-source software applications such as mobile applications and web applications.

SEN 407: Software Engineering Professional Practice

Accreditation, certification and licensing, codes of ethics and professional conduct, nature and role of professional societies and software engineering standards, economic impact of software, employment contract, legal issues, documentation, and trade-off analysis; Group dynamics and psychology-dynamics of working in teams / groups, individual cognition, dealing with problem complexity, interacting with stakeholders, dealing with uncertainty and ambiguity, dealing with multi-cultural environment; Communication skills-reading, understanding and summarizing, writing, teams and group communication and presentation skills.

SEN 408: Distributed, Parallel and Cloud Computing

Analysis and design of parallel and distributed algorithms, language and OS for parallel processing, GPU computing, Architecture of parallel systems, tools for parallel computing, parallel and distributed database, Networking aspect of parallel and distributed computing, High performance computing, multimedia application of parallel and distributed computing, Distributed file system, cloud computing, cloud as a service, infrastructures, management and operations, performance, scalability, reliability, virtualization, performance evaluation.

SEN 409: Software Engineering Security

History and terminology, security mindset, design principles, system / security life cycles, security implementation mechanisms, information assurance analysis model, disaster recovery, security mechanism, cryptography, authentication, redundancy and intrusion detection. Operational issues trends, auditing, cost / benefits analysis, asset management, standards, legal issues, policy creation, attacks and its types, Forensic legal systems, digital forensics.

SEN 410: Software Architecture and Design

An in-depth look at software design. Continuation of the study of design patterns, frameworks, and architectures. Survey of current middleware architectures. Design of distributed systems using middleware. Component based design. Measurement theory and appropriate use of 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 engineering.

Electives

SEN 411: Artificial Intelligence

This course provides an introduction to the field of artificial intelligence. Topics include knowledge acquisition, knowledge representation, knowledge-based search techniques, machine reasoning and learning. Emphasis will be put on algorithms for search, inference, constraint satisfaction and optimization. Applications in tasks such as expert systems, data mining, game playing, natural language understanding, computer vision, speech recognition, robotics and other knowledge intensive problems requiring smart agents will be examined. Functional programming using a LISP-based language and/or logic programming with Prolog will be used to illustrate basic data structures for knowledge representation, reasoning and guided inference using AI programming techniques. Programmes that implement problem-solving search methods, logical reasoning techniques, production-rule systems, and neural networks will be studied and developed. Students are expected to complete a number of written and programming assignments as well as an individual semester project.

SEN 412: Special Topics in Software Engineering

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

SEN 413: Engineering Mobile Applications

Introduction to developing mobile applications, beginning with mobile operating systems 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 and SQL database engine and multithreading.

SEN 414: Fault Tolerant Computing

Introduction and overview of fault tolerant schemes, fault and error modeling, test generation and fault simulation, concept in fault tolerance, reliability / availability modeling, system level diagnosis, low level fault tolerance, coding techniques (e.g., Hamming code etc.), high level fault tolerant technique in system, rollback, check pointing, reconfiguration, software fault tolerance, fault tolerant routing.

SEN 415: Game Design and Development

This 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, Students will play games, analyze them, and complete portion of game designs with appropriate documentation.

SEN 416: Modeling and Computer Simulation

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 report and presentations, training simulators.