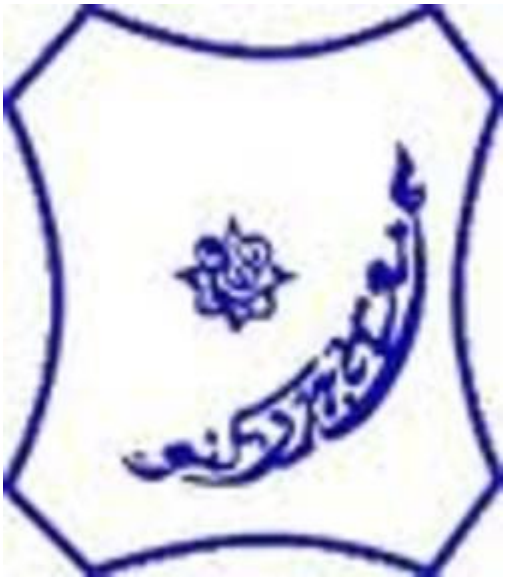
BAYERO UNIVERSITY, KANO

DEPARTMENT OF SOFTWARE ENGINEERING

FACULTY OF COMPUTER SCIENCES AND INFORMATION TECHNOLOGY

UNDERGRADUATE STUDENTS HANDBOOK

THE UNIVERSITY CREST



The Crescent and the Star

The Crescent: **Jamiatu Bayero Kano**

(Symbol & Unit Time) Bayero University Kano

The Star: **Motto: "**Wa Fawqa Kulli Dhi Ilmin Alim**"**

(Guiding Light) "….But overall endowed with knowledge is one,
All knowing"

PHILOSOPHY AND CARDINAL PRINCIPLES OF THE UNIVERSITY

Abdullahi Bayero College was reconstituted into a University College with effect from October 01, 1975 and into Bayero University in October 01, 1977.

The Academic Development Committee outlined a philosophy consistent with the traditions and character, which the College acquired during its years of existence. Briefly, it is that the University is to pay due respect to human values. It is to: "be built on those beliefs, values and traditions that the society holds most sacred". The University should strive to be symbol of the spirit of its community, the guardian of its morals, the formulator of its hopes, be they spiritual or material. It should monitor the inevitable changes that come with time and, in assimilating these essential values and nuances, the University, recognizes the need to:

1. Deepen individuals' awareness of the value of the cultural heritage;
2. Foster and reinforce the moral, ethical and spiritual values sacred to the culture;
3. Provide the knowledge, skills and technology required for the benefit of the wider community;
4. Contribute, by research, to the universal body of knowledge;
5. Further the objectives of a united, fair and just community, society and nation"

In pursuing its aims and objectives, the University will be constantly guided by the requirements of the national plans, manpower needs and sheer physical and environmental factor.

1.0. INTRODUCTION

The discipline of Software Engineering focuses on producing graduates who are ready to develop and maintain quality software systems of scale for organizations and businesses within the constraints of time, budget and other requirements. In addition to its core computer science technical foundation, Software Engineering 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 Software Engineering specific curriculum must be able to develop students into software engineers through course work, practicals and applied project experience.

1.1. BRIEF HISTORY OF THE DEPARTMENT OF SOFTWARE ENGINEERING

The history of the Department can be traced to 1900's when the Computer Center was established in the University. The Center functioned as a service unit providing computing facilities to the entire university community. Lecturers and students from different faculties, but mainly those from the Faculty of Science, had to learn programming on their own in order to use the facilities.

As the Information Technology revolution rapidly unfolded and new computer applications evolved in different areas, it soon became evident that there was the need to have specialized computer professionals that were especially fluent in algorithm design, systems analysis, programming and software development. The University therefore started a B.Sc. program in Computer Science in 2001 domiciled in the Department of Mathematics.

Successful as this program was in annually producing hundreds of Computer Science graduates, with the exponential increase in the demand for more computer professionals in both quantity and specialization, it became obvious that this was not enough. The

University set up a high powered committee to look into this and the major recommendation was to establish a Faculty of Computer Sciences and Information Technology with three (3) departments namely Computer Science, Information Technology and Software Engineering.

The new Faculty and the Departments took off in 2012 but only the B.Sc. program in Computer Science, which was transferred from the Department of Mathematical Sciences, had due permission from the National Universities Commission (NUC) to enroll students. Thus, the newly established Department of Software Engineering, which barely had six (6) members of staff, could only assist in teaching Software Engineering courses to Computer Science students. In 2017, NUC finally gave the consent to the University to start the B.Sc. program in Software Engineering.

1.2. 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 programmes in Software Engineering 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 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 organizational behavior and management in the software development life-cycle.

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

(v) Nurturing of partnership between the institution and the software industry for 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 ecosystem while protecting human health and welfare and the 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 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. 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:

(i) To appreciate the importance of computer science (as a base) in such areas 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 systems in which software plays a critical role.

(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.

1.4. LEARNING OUTCOMES:

(a) Regime of Subject Knowledge

(i) Have a thorough knowledge of scientific and engineering practice and theory in computing and be able to extend this knowledge through self-led study

(ii) Understand the impact of globalization on computing and software engineering

(iii) Recognize the challenges and opportunities for the software industry in Nigeria and the region.

(iv) Identify requirements for specialised computing systems and propose solutions to fulfil them

(v) Use and, where appropriate, modify for specific use, established systems development methods

(vi) Explain the relationships between computer systems and other natural and artificial systems in the modern world at appropriate levels of abstraction

(vii) Explain the concepts of computer programming and critically evaluate and predict their utility in models, tools and applications

(viii) Demonstrate advanced, specialist theoretical and practical knowledge in a range of computer science sub-fields

(ix) Explain legal issues relating to computing: intellectual property, data protection, computer misuse and health and safety

(x) Explain the principles and practice of software engineering in a modern industrial context

(b) **Competences and Skills**

(i) Analyse and abstract problems and propose and apply effective solutions

(ii) 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.

(iii) Participate productively on software project teams involving others from a variety of disciplines and backgrounds

(iv) Develop and critically evaluate specifications for specialist computer systems

(v) Elicit, analyze and specify software requirements through a productive working relationship with project stakeholders

(vi) Apply software engineering best practice to the development of computer systems

(vii) Demonstrate software engineering application domain knowledge having engineered a software product of value to a specific organization or to address a societal need.

(viii) Implement specialist computer systems from given specifications

(ix) Plan and manage large scale projects

(x) Communicate requirements and proposals for computer systems to other computing professionals

(xi) Communicate effectively through oral and written reports, and software documentation

(xii) Evaluate the impact of potential solutions to software engineering problems in a global society, using their knowledge of contemporary issues

(xiii) Employ an entrepreneurial approach to provide sustainable software engineering solutions.

(xiv) Identify the common needs of industry from computer systems and apply controlled compromise in meeting requirements

(xv) Design and execute methodologically sound scientific and engineering studies

(xvi) Demonstrate advanced specialist skills in addressing the problems of Computer Science and its sub-fields

(xvii) Understand, evaluate, synthesize and apply complex ideas

(c) **Behavioral Attributes**

(i) Apply appropriate codes of ethics and professional conduct to the solution of software engineering problems

(ii) Explain the issues of professionalism in computing including the need for continuing professional development

(iii) Assess the nature of intellectual property and its ownership, and respect it accordingly

(iv) Identify resources for determining legal and ethical practices in other countries as they apply to computing and software engineering

1.5. BASIC ADMISSION REQUIREMENTS

Candidates are admitted into the Software Engineering degree programme in any of the following two ways: (i) the Unified Tertiary Matriculation Examination (UTME); (ii) Direct Entry;

1. Unified Tertiary Matriculation Examination (UTME) mode

(a) UTME subjects must include English Language, Mathematics, Physics and one other acceptable science subject;

(b) Five Credit level passes O’L Level in nationally recognised examination at not more than **TWO** sittings, which must include English Language, Mathematics, Physics, and two other acceptable science subjects.
It is also desirable for candidates to pass Computer Studies and Further Mathematics at credit level, such candidates shall have added advantage.

2. Direct Entry (DE) Mode

For Direct Entry, in addition to the minimum requirements in 1(b), candidates with:

(a) Two A' Level passes with minimum of C grade in Mathematics and Physics are eligible for consideration for admission into 200 level;

(b) At least Upper Credit pass in National Diploma/Higher National Diploma in Computer Science/Engineering or related disciplines from a recognised Polytechnic or its equivalent are eligible for consideration for admission into 200 level.

1.6. PROGRAMME STRUCTURE AND DURATION

The Department of Software Engineering offers a four (4) year degree Programme for UTME candidates and a three (3) years Programme for DE candidates. There are two (2) semesters of formal University studies in each Academic Session. At 300 level, a Student is expected to go for 6 months Students Industrials Work Experience Scheme (SIWES), after completion of the first semester courses at the end of which a report about what the Student learned in the industry has to be written, presented and defended. At 400 level, Students undertake a one year project in any field of interest in Software Engineering besides the usual prescribed courses.

(i) Programme Duration

A Student in Software Engineering shall normally complete the Programme in eight (8) semesters, but may be allowed to continue for a total of twelve (12) semesters and still qualify for a degree provided the Students CGPA is 1.00 or above.

(ii) Semester Duration

The minimum during for teaching in a semester is fifteen (15) weeks. Excluding public holidays and semester breaks. A duration of two (2) to three (3) weeks is usually reserved for examinations after the teaching period.

1.7. REQUIREMENTS FOR GRADUATION

To be eligible for the award of B.Sc. degree in Software Engineering, a student must have:

(i) Passed all the core courses, university and faculty/school required courses and electives;

(ii) Accumulated a minimum of 120 course units for students admitted through UTME and 90 for students admitted to 200 level Direct Entry.

(iii) Attain a minimum CGPA of 1.50.

1.8. COURSE NOMENCLATURE

The coding of course codes in Software Engineering Department is structured using alphanumeric format described as follows:

(i) A three uppercase prefix "SWE" derived from the term Software Engineering or "CSC" derived from Computer Science or "ITC" derived from Information Technology or "CBS" derived from Cyber Security.

(ii) A four digit code (example 1301) whose first digit signifies the level at which the course is offered, the second digit signifies the credit value of the course and the two last digits are used for identification purposes.

Note that the credit units of a course determines the hours of lectures for that week. For example (SWE1301) implies 3 hours of lectures per week.

1.9. COURSE SYSTEM

The following standard terminologies are used for different categories of courses in the Department:

(i) **Core/Compulsory Course:**

A course which every student must compulsorily take and pass in any particular programme at a particular level of study.

(ii) **Elective Course**

A course that students take within or outside the faculty. Students may graduate without passing the course provided the minimum credit unit for the course had been attained.

(iii) **Optional Course**

A course which students can take based on interest and may count towards the minimum credit unit required for graduation.

1.10. COURSE ASSESSMENT

Each course is assessed by continuous assessment and an end- of-semester examination. The continuous assessment normally accounts for 30-40%, while the end-of-semester. Examination accounts for 60-70% of the overall marks for each course.

1.11. REGISTRATION

Each Student must register and the pay the appropriate fees at the beginning of each session. A fresh Student must complete the registration process before the closure of central registration of new Students. Failure to complete the process within this time shall attract late registration fees charges or forfeiture of the admission. A returning Student who fails to complete the registration process with the specific period approved for registration shall pay a late registration fees as may be prescribed by the University. A returning Student who fails to register within four (4) weeks of commencement of the registration exercise shall not be allowed to register. Such a Student shall be deemed to have withdrawn, unless (s) he provides a reason acceptable to the Senate, in such case (s) he can be considered for suspension of studies.

i. Credit Load

A Student shall normally in any semester be required to register a minimum of 15 credits and a maximum 24 credits. This implies that for an Academic Session a Student shall register for a minimum of 30 credits and maximum of 48 credits. There is however exceptional circumstances such as:

a. Student on SIWES where such an exercise lasts for a semester and its credit load is less 15 and/or where the sessional credit load is less than 30 credits
b. Spill-over Students requiring less than 15 credits in a semester and/or less than 30 credits in a session to graduate.

ii. Registering for Courses

a. Courses are to be registered for by students sequentially. Thus, a Student must register for Level II courses before registering for Level III. Moreover, when registering, a Student shall first enter lower level courses (failed, or not taken) before higher level course.

b. If a Student fails a required course, s(he) must register for it as “carry over” in all subsequent sessions until the course is cleared except where in level 300 second semester where Students go for SIWES.

iii. The Add/Drop Procedure

Registered Students may make minor changes in the courses registered for by adding and/or reducing some courses at the beginning of the second semester. The Student shall discuss the proposed changes with his/her Level Coordinator to ensure changes are in order. The Student must also submit to the Level Coordinator the updated copies of the documents.

2.0. EXAMINATION

A Student shall take an examination at the end of each semester. Thus, in an academic session a Student shall sit for two (2) examination.

2.1. ELIGIBILITY FOR EXAMINATION

In order to be admitted into an examination, a Student must have registered the course.
A Student must have at least a 75% attendance.
Note: The Department follows Examination Rules prescribed in the University handbook.

2.2. EXAMINATION INSTRUCTIONS FOR STUDENTS

(i) A student is responsible to acquaint himself/herself with the date and timing and venue of the examination.

(ii) Students MUST come along with their Identity Card in all the examinations.

(iii) Students shall be at the examination venue at least 30 minutes before the start of the examination.

(iv) Students are advised to strictly adhere to the Examination Rules and Regulation in the University handbook.

(v) All sick cases during the examination must be report promptly to the Level Coordinator.

2.3. GRADING OF COURSES

Grading of courses shall be done by a combination of percentage marks and letter grades translated into a graduated system of Grade Point as shown in Table **1.**

Table 1: Grade Point System

Mark %	Letter Grade	Grade Point
70 – 100	A	5
60 – 69	B	4
50 – 59	C	3
45 – 49	D	2
40 – 45	E	1
< 40	F	0

(i) 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 product of the total number of Units x Grade Point (TUGP) by the total number of units (TNU) for all the courses taken (whether passed or failed) in the semester as illustrated in Table **2**. 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.

Table 2: Calculation of GPA or CGPA

Course	**Credit Unit**	**Grade Point**	**Units x Grade Point (UGP)**
C_1	U_1	GP_1	U_1 x GP_1
C_2	U_2	GP_2	U_2 x GP_2
-	-	-	-
-	-	-	-
C_i	U_i	GP_i	U_i x GP_i
-	-	-	-
-	-	-	-
C_N	U_N	GP_N	U_N x GP_N
TOTAL	**TNU**		**TUGP**

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

Table 3: GPA Example

Course	Credit	Grade Letter	Grade Value	Grade Points
First Semester				
CSC1303	3	A	5	3*5=15
ITC1203	2	C	3	2*3= 6
SWE1301	3	B	4	3*4=12
MTH1301	3	B	5	3*5=15
PHY1210	2	A	5	2*5=10
GSP1201	2	F	0	2*0=0
	15			58
GPA= (58/15) = 3.86				

Table 4: CGPA Example

Course	Credit	Grade Letter	Grade Value	Grade Points
First Semester				
CSC1303	3	A	5	15
ITC1203	2	B	4	8
SWE1301	3	B	4	12
MTH1301	3	B	5	15
PHY1210	2	A	5	10
GSP1201	2	C	3	6
	15			66
GPA = (66/15) = 4.40				
Second Semester				
CBS1202	2	B	4	8
SWE1304	3	C	3	9
CST1301	3	A	5	15
	8			32
GPA = (32/8) = 4.00				
CGPA = (98/23) = 4.26				

(ii) Absent (ABS) and Incomplete Grade

a. ABS: A Student shall be awarded ABS if s (he) did not sit for an Examination in a registered course. ABS is computed as F grade in GPA/CGPA calculations**.** A Student is allowed to verify an ABS grade if s (he) has sat for the Examination.

b. Incomplete Grade (I) means the assessment of a course the formal examination of which is not written under the permission of Senate. This means that the course is not included in GPA/CGPA calculation even though the Student has registered for it

(iii) Degree Classifications

The determination of the class of degree shall be based on the Cumulative Grade Point Average (CGPA) earned at the end of the programme. The CGPA shall be used in the determination of the class of degree as summarized in Table **5.** It is important to note that the CGPA shall be calculated and expressed correct to two decimal places.

Table 5: Degree Classification

CGPA	**Class of Degree**
4.50 - 5.00	First Class Honours
3.50 - 4.49	Upper Second Class Honours
2.40 - 3.49	Lower Second Class Honours
1.50 - 2.39	Third Class Honours
1.00 - 1.49	Pass

(iv) Probation

A student whose Cumulative Grade Point Average is below 1.50 at the end of a particular year of study, earns a period of probation for one academic session. A student on probation is allowed to register for courses at the next higher level in addition to his/her probation level courses provided that:

i) the regulation in respect of student work-load is complied with; and

ii) the pre-requisite courses for the higher level courses have been passed.

(v) Withdrawal

A candidate whose Cumulative Grade Point Average is below 1.50 at the end of a particular year of probation should be required to withdraw from the University. Withdrawn Students are free to reapply for fresh admission into the University using the usual UTME channel. If readmitted, such students cannot use any credits earned in the previous programmes from which they were withdrawn.

2.4. PUNISHMENTS FOR EXAMINATION MISCONDUCT

(i) Expulsion: This means terminating the student's programme of study. An expelled Student is forever disqualified from admission into any programme in the University. If s(he) applies and is admitted due to oversight, Senate would expel him/her again as soon as the case is discovered. Senate shall withdraw any qualification conferred on any previous expelled Student.

(ii) Rustication: For a non-final Student this means staying away from academic work for the period indicated by Senate. For a final year student, "Rustication" means cancelling the work of the semester(s) in which the offence occurred and registering afresh in a new semester/session. As rustication is a punishment, the period of rustication counts in determining the Students maximum period of stay in the University.

2.5. LEAVE ENTITLED TO STUDENTS

(i) Maternity Leave

A female Student must submit a medical certificate showing the expected date of delivery (EDD), not less than 6 weeks in advance of the date, and can be granted dispensation from course work for a period of 12 weeks, commencing 6 weeks before the (EDD). Incomplete grade will normally be awarded only if the examination occurs within a period beginning 6 weeks before the EDD and ending 9 weeks after the actual date of delivery. A faculty Board may at its discretion extend this period on receipt of a medical statement indicating exceptional circumstances. If the EDD interferes with a substantial part of the semester’s coursework, the Department shall recommend suspension of studies for the Student.

a. Suspension of Studies means a period approved by Senate (based on an application by the Student with appropriate supporting documents) for the Student to be away from the University. It is given for a maximum of one session in the first instance, but could be renewed (on application) for one more session only. The period of suspension of studies does not count in determining the maximum period pf stay at the University by the concerned Student.

(ii) Sick Leave

A Student may be granted dispensation from course work for a period of time if s (he) get sick. Incomplete grade will normally be awarded only if the examination occurs within the period. If the sickness interferes with a substantial part of the semester’s coursework, the Department shall recommend suspension of studies for the Student. A student must however apply and submit the necessary documents on time.

3.0. COURSE OUTLINE

100 LEVEL

S/N	Status	Code	Title	Credits
			First Semester	
1	Core	SWE 1301	Introduction to Problem Solving and Software Dev.	3
2	Core	CSC 1303	Introduction to Computer Science	3
3	Core	CST 1311	Introduction to Probability	3
4	Core	ITC 1203	Fundamentals of IT	2
5	Core	MTH 1301	Elementary Mathematics I	3
6	Core	PHY 1210	Mechanics	2
7	Core	PHY 1220	Electricity and Magnetism	2
8	Core	GSP 1201	Use of English	2
			TOTAL	20
Second semester				
9	Core	SWE 1304	Introduction to Computer Programming	3
10	Core	CST 1301	Introduction to Application Packages	3

11	Core	CBS1202	Fundamentals of Cyber Security	2
12	Core	MTH 1303	Elementary Mathematics II	3
13	Core	PHY1230	Behavior of Matter	2
14	Core	GSP1202	Use of Library, Study Skills and ICT	2
			TOTAL	**15**
			LEVEL I TOTAL	**35**

200 LEVEL

S/N	Status	Code	Title	Credits
First Semester				
1	Core	SWE2301	Introduction to Software Engineering	3
2	Core	CSC2323	Discrete Structures	3
3	Core	CSC2253	Data Structures and Algorithms	2
4	Core	CSC2211	Computer System Theory	2
5	Core	ITC2201	Introduction to Web Computing	2
6	Core	GSP2204	Foundation of Nigerian Culture, Government and Economy	2
7	Core	GSP2206	Peace Studies and Conflict	2
8	Core (DE only)	GSP2201	Use of English	2
			TOTAL	**16(18)**
Second semester				
9	Core	SWE2204	Human Computer Interaction	2
10	Core	SWE2210	Software Construction	2
11	Core	SWE2211	Software Engineering Process	2
12	Core	SWE2314	Computer Programming II	3
13	Core	SWE2315	System Analysis and Design	3
14	Core	CSC2210	Operating System I	3
15	Core	CSC2206	Computer Architecture and Organization I	2
16	Core	CSC2204	Analysis of Algorithms	2
17	Core	MTH2205	Linear Algebra I	2
18	Core	GSP2205	Logic and Philosophy	2
19	Core	GSP2202	Use of Library, Study Skill &ICTs	

	(DE Only)			
			TOTAL	**23(25)**
			LEVEL II TOTAL	**39(43)**

300 LEVEL

S/N	Status	Code	Title	Credits
			First Semester	
1.	Core	SWE3311	Object-Oriented Analysis and Design	3
2.	Core	SWE3202	Software Testing and Quality	2
3	Core	ITC3303	Data Communication and Networking	3
4.	Core	ITC3304	Web Application Development	3
5.	Core	ITC3305	Database Systems	3
6.	Core	CSC3343	Operating System	3
7.	Core	CSC3307	Computational Science and Numerical Methods	3
8.	Core	CSC3251	Research Methodology	2
9	Elective	CBS3209	Fault Tolerant Computing	2
			Total	**22(24)**
			Second Semester	
10	Core	SWE3600	SIWES	6
			TOTAL	**6**
			LEVEL III TOTAL	**28(30)**

400 LEVEL

S/N	Status	Code	Title	Credit
			First Semester	
1	Core	SWE4201	Software Configuration Management and Maintenance	2
2	Core	SWE4203	Software Engineering Project Management	2
3	Core	SWE4207	Software Engineering Professional Practice	2
4	Core	SWE4209	Concepts of Programming Languages	2

5	Core	SWE4211	Software Engineering Security	2
6	Core	SWE4210	Software Architecture and Design	2
7	Core	CSC4303	Operation Research I	3
8	Core	EEP3201	Introduction to Entrepreneurial Skills	2
9	Elective	SWE4215	Embedded Systems	2
10	Elective	SWE4218	Modelling and Computer Simulation	2
			Total	17(21)
Second Semester				
10	Core	SWE4202	Software Engineering Economics	2
11	Core	SWE4206	Open Source Software Development and Applications	2
12	Core	SWE4208	Distributed, Parallel and Cloud Computing	2
13	Core	SWE4600	Final Year Student’s Project	6
14	Core	EEP4201	Business Creation and Growth	2
15	Elective	SWE4212	Special Topics in Software Engineering	2
16	Elective	SWE4216	Game Design and Development	2
			TOTAL	14(18)
			LEVEL IV TOTAL	29(37)

*A student MUST register one of the elective in each semester

3.1. COURSE DESCRIPTIONS

SWE1301: Introduction to Problem Solving and Software Development

Problem solving strategies, Role of algorithm in problem solving process, implementations strategies, concepts and properties of algorithm, development of flow chats, pseudo codes, program objects implementation of algorithm in a programming language, Introduction to Software Engineering, Software Development Life Cycle Model.

CSC1303: Introduction to Computer Science

Introduction to Computers; the World Wide Web; Application Software; Computer Components; Input-output-storage devices; Operating System and utility programs; Communication and Networks; Database Management System; Computer Security, Safety, Ethics and Privacy.

CST1311: Introduction to Probability

Measure of location and dispersion in simple and grouped data exponential. Elements of probability and probability distribution, normal, binomial, poison, geometric, Negative binomial distributions. Estimation and tests of hypothesis concerning the parameters of distribution. Regression, correlation, and analysis of variance contingency table Non-parametric inference.

ITC1203: Fundamentals of IT

Pervasive themes in information technology, information technology system model, a gentle introduction to information technology, human-computer interaction, Information management, networking, platform technologies, programming and web systems and technologies and its related and information disciplines, information technology application domains.

MTH1301: Elementary Mathematics

Elementary set theory, subsets, union, intersection, complements, Venn diagrams. Real numbers; integers, rational and irrational numbers, mathematic I, induction real sequences and series, theory of quadratic equations, binomial theorem. Complex numbers; algebra of complex numbers; the Argand Diagram. Re Moivre's theorem, nth roots of unity. Circular measure, trigonometric functions of angles of any magnitude, addition and factor formulae.

PHY1210: Mechanics

Space and Time, Units and dimension, Kinematics; Fundamental Laws of Mechanics, statics and dynamics; work and energy; Conservation laws. 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, Applications.

PHY1220: Electricity and Magnetism

Electrostatics; conductors and currents; dielectrics; magnetic fields and induction; Maxwell's equations; electromagnetic oscillations and waves; Applications.

GSP1201: Use of English

Effective communication and writing in English, Language skills, writing of essay answers, Comprehension, Sentence construction, Outlines and paragraphs, Collection and organization of materials and logical presentation, Punctuation.

SWE**1304: Introduction to Computer Programming**

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 course.

CST1301: Introduction to Application Packages

Functions, purpose and applications of spreadsheet software, common spreadsheet concept, data entry and organization, spreadsheet formulae, numeric and formula entries formatting data, what-if analysts, spreadsheet graphics spreadsheet base capabilities. MICROSOFT EXCEL & QJATRO PRO are to be used for this course.

CBS1202: Fundamentals of Cybersecurity

Provides an over view of the introductory topics in cyber security, topics includes basic concept, confidentiality, integrity, availability, authentication, access control, non-repudiation and fault tolerant methodologies for implementing security, securities policy, best current practices, testing security and incident report, risk management disaster recovery, access control, basic cryptography and software application vulnerabilities.

MTH1303 Elementary Mathematics II

Calculus: Function of a real variable, graphs, limits and idea of continuity. The derivative, as limit of rate of change. Techniques of differentiation. Extreme curve sketching;

Integration as an inverse of differentiation. Methods of integration, Definite integrals. Application to areas, volumes.

PHY1230: Behavior of Matter

Electrostatics; conductors and currents: dielectrics: magnetic fields and induction: Maxwell equation: electromagnetic oscillations and wave application.

GSP1202: Use of Library, Study Skills and Information Communication Technology (ICT)

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.

SWE2301: Introduction to Software Engineering

Software engineering concepts and principles; design, development and testing of software systems. Introduction to software life cycle. 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 a team-project including user interface considerations.

CSC2323: Discrete Structures

This covers the Mathematics needed for Computer Science. Topics covered include: functions, relations, propositional and first order predicate logic, set theory, proofs and their construction, counting and elementary probability.

CSC2253: Data Structures and Algorithms

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), 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. Introduction to algorithm for parallel & distributed computing.

CSC2212: Computer System Theory

Topics include valid and invalid arguments, translating from English to the language of proportional and predicate logic, formal deduction and its role in providing the validity of an argument: logic and computer science, how to build a circuit from logic gates and how to minimize circuits using propositional logic, introduction to prolog, a programming language based on logic and the application of logic in computer science. AI automated theorem- provers,, expert system etc. Fundamental concepts of computer systems and system programming. Hardware fundamental including digital logic, memory systems, processor design, buses, I/O subsystems data representations, computer arithmetic, microprogramming, and instruction-set, architecture. Software concepts including assembly language programming, operating systems, assemblers, linkers and compilers, computer peripherals interfacing and maintenance.

ITC2201: Introduction to Web Computing

Introduction to Web Computing An introduction to the Internet, the World Wide Web, and web development students will create interactive web pages by writing HT 'and and CSS and by programming in JavaScript- Topics include the origins of the web, the roles and operations of web browsers and web servers, interacting with web applications through forms, and using digital media. style sheets to separate document structure and document formatting;

GSP2204: Foundation of Nigerian Culture, Government, & Economy

Study of Nigerian history, culture and arts in pre-colonial times, Nigerian’s perception of his world, Culture areas of 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.

GSP2206: Peace Studies & Conflict Resolution

Basic Concepts in peace studies and conflict resolution, Peace as vehicle of unity and development, Conflict issues, Types of conflict, e. g. Ethnic/religious/political/ economic conflicts, Root causes of conflicts and violence in Africa, Indigene/settler phenomenon, Peace – building, Management of conflict and security. Elements of peace studies and conflict resolution, developing a culture of peace, Peace mediation and peace-keeping, Alternative Dispute Resolution (ADR). Dialogue/arbitration in conflict resolution, Role of international organizations in conflict resolution, e.g. ECOWAS, African Union, United Nations, etc.

GSP2201: Use of English

Effective communication and writing in English, Language skills, writing of essay answers, Comprehension, Sentence construction, Outlines and paragraphs, Collection and organization of materials and logical presentation, Punctuation.

SWE2204: Human Computer Interaction

Introduces the principles of user interface development, focusing on three key areas: (1). 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.

SWE2210: Software Construction

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.

SWE2211: Software Engineering Process

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.

SWE2314: Computer Programming II

Builds on CST1304 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.

SWE2315: Software Requirements Analysis and Specification

Definition of a software requirement, product and process requirements, functional and non–functional requirements, emergent properties, quantifiable requirements, and system and software requirements; Requirements process – process models and actors, process support and management, and process quality and improvement; Requirements elicitation – requirements sources and elicitation techniques.

CSC2210: Operating System I

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. Prerequisites.

SWE2206 – Computer Organisation & Architecture I

Differences between computer architecture and computer organisation, basic structure of computers; performance evaluation: metrics and calculations, performance equations, Amdahl's law; CPU organization and micro-architectural level design; Instruction set design; register transfer; RISC design principles; data-path design; controller design; memory system; addressing; microprogramming; computer arithmetic; 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.

CSC2204: Analysis of Algorithms

Introduction to algorithms and its 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.

MTH2205: Linear Algebra I

Vector spaces 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 transformations and matrices. Algebra of matrices.

GSP2205: Philosophy and Logic

A brief survey of the main branches of Philosophy Symbolic Logic Special symbols in symbolic Logic-conjunction, negation, affirmation, disjunction, equivalent and conditional statements 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; Distinction between inductive and deductive inferences; etc. (Illustrations will be taken from familiar texts, Including literature materials, Novels, Law reports and newspaper publications).

GSP2202: Use of Library, Study Skills & ICTs

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.

SWE3311: Object-Oriented Analysis and Design

Object–oriented approach to information system development, particularly in reference to the earlier stages of analysis and design. Importance of modelling, principles of modelling, object–oriented modelling, conceptual model of the Unified Modelling Language (UML), architecture, software development life cycle. The principles and basic concepts of object orientation and the different aspects of object–oriented modelling as represented by the UML technique. Case study of a typical UML–based CASE tool.

SWE3203: Software Testing & 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.

ITC3303: Data Communications and Networking

Overview of data communication, communication models and networking, analog and digital data transmission, transmission impairment, channel capacity, synchronous and asynchronous transmission, error detection and correction, flow control and error control. Multiplexing and DE multiplexing techniques (FDM, STDM, ADSL and xDSL) Ethernet interface, IEEE 802.3 and IEEE 802.11 MAC layer, interface standards (RS-232, RS-449 and X.21) packet switching, frame relay and ATM switching, bridges, layer 2, and layer 3 switches.

ITC3304: Web Application Development

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

ITC3305: Database Systems

The course will cover the concept, principles, components, development and application of database systems. The conceptual models and structures necessary to 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.

CSC3343: Operating System

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. Prerequisites.

CSC3307: Computational Science and Numerical Methods

Solution of algebraic and transcendental equation, curve fitting errors analysis, interpolation and approximation, zero of non-linear equation of one variable, system of linear equation, numerical differentiation and integration: integration equation numerical solutions of ODE: initial value problems.

CSC3251: Research Methods

Foundation of research, problem identification and formulation, research design, qualitative and quantitative research, measurement, sampling data analysis, interpretation of data and technical report writing, use of encyclopedia, research guides handbook etc, academic data bases for computing disciplines, use of tools for research, reference management software, software for detection of plagiarism.

CBS3209: Fault Tolerant Computing

Introduction and overview of faulty tolerant scheme, full and error modelling, test generation and fault simulation concept in faulty tolerance, reliability availability modelling system diagnosis low level fault tolerance- codding tecgniques(basic modelling; system level diagnosis: low level fault tolerance- code techniques. (Basic principles, parity bit codes, hamming codesm error detection and retransmission codes, burst error correction codes, reed Solomon codes etc.)high level fault tolerance technique in system, roll back, check pointing, reconfiguration, software fault tolerance, fault tolerant technique, integrated software/hardware fault tolerance, redundancy, parallel redundancy, spares and repairs, apportionment, system versus component redundancy parallel redundancy, RAID system reliability, N modular redundancy, software reliability and recovery techniques, network system

SWE3600: SIWES

Students are attached to private and public organizations for a period of six months with a view to making them acquire practical experience and to the extent possible, develop skills in all areas of computing. Students are supervised during the training period and shall be expected to keep records designed for the purpose of monitoring their performance. They are also expected to submit a report on the experience gained and defend their reports.

SWE4201: Software Configuration Management & Maintenance

Management of the software configuration management process – organisation 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, and software maintenance measurement; Maintenance process – maintenance processes and activities; Techniques for maintenance – program comprehension, reengineering, reverse engineering, migration, and retirement;

SWE4203 – Software Engineering Project Management

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 of plans, software acquisition and supplier contract management, implementation of measurement process, monitor process, control process, and reporting; Review and evaluation – determining satisfaction of requirements, and reviewing and evaluating performance; Closure – determining closure and closure activities; Software engineering measurement – establish and sustain measurement commitment, plan the measurement process, preform the measurement process; and evaluate measurement; Software engineering management tools.

SWE4207: 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 contracts, 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, and dealing with multicultural environments; Communication skills – reading, understanding and summarising, writing, teams and group communication, and presentation skills.

SWE4209: Concepts of Programming Languages

Preliminaries, 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.

SWE4210 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.

SWE4211: Software Engineering Security

History and terminology, security mindset, design principles, system/security life–cycle, security implementation mechanisms, information assurance analysis model, disaster recovery, and forensics; Security mechanisms–cryptography, authentication, redundancy, and intrusion detection; Operational issues–trends, auditing, cost/benefit analysis, asset management, standards, enforcement, legal issues, and disaster recovery; Policy–creation of 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

CSC4303: Operations Research

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.

EEP3201: Introduction to Entrepreneurial Skills

Introduction to entrepreneurship and new venture creation; Entrepreneurship in theory and practice; The opportunity, Forms of business, Staffing, Marketing and the new venture; Determining capital requirements, raising capital; Financial planning and management; Starting a new business, Feasibility studies; Innovation; Legal Issues; Insurance and environmental considerations. Possible business opportunities in Nigeria’s, nails, screws making Dyeing/Textile blocks paste making.

SWE4215: Embedded Systems

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.

SWE4218: Modelling and Computer Simulation

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

SWE4202: Software Engineering Economics

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 analysis, minimum acceptable rate of return, return on investment and capital employed, cost–benefit analysis, cost–effectiveness analysis, break–even analysis, business case, multiple attribute evaluation, and optimisation analysis; Practical considerations – the “good enough” principle, friction – free economy, ecosystems, and offshoring and outsourcing.

SWE4206: Open Source Development & Applications

Introduces concepts, principles and applications of open source software. Discusses about open source software development process. Covers economy, business, societal and intellectual property aspects of open source software. Obtain hands-on experiences 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.

SWE4208: Distributed, Parallel and Cloud Computing

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 systems, 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 Systems; 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.

SWE4600: Final Year Student Project

An independent or group investigation of appropriate software, hardware, communication and networks or IT related problems in Software Engineering carried out under the supervision of a lecturer. Before registering, the student must submit a written proposal to the supervisor to review. The proposal should give a brief outline of the project, estimated schedule of completion, and computer resources needed. A formal written report is essential and an oral presentation may also be required.

EEP4201: Business Creation and Growth

Some of the ventures to be focused upon include the following: Soap/Detergent, Tooth brushes and Tooth paste making, Photography, Bricks, Rope making, Plumbing, Vulcanizing, Brewing Glassware production/Ceramic, production, Paper production, Water treatment/Conditioning/Packaging, Food processing/packaging/preservation, Metal working/Fabrication – Steel and aluminum door and windows, Training industry, Vegetable oil/and Salt extractions, Fisheries/Aquaculture, Refrigeration/Air conditioning, Plastic making, Farming (crop), Domestic Electrical wiring, Radio/TV repairs, Carving, Weaving, Brick laying/making, Bakery, Tailoring.

SWE4212: 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 or guests, students are expected to do substantial readings on their own.

SWE4216: Game Design and Development

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