FEDERAL UNIVERSITY OF TECHNOLOGY, MINNA
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE

UNDERGRADUATE ACADEMIC BRIEF
2026 – 2030

FORWARD

Computer science is a dynamic field of study albeit, with some fundamental and foundational courses which form the basis of the discipline. In light of this, this reviewed curriculum is prepared to ensure consistencies with the Nigeria Universities Commission Benchmark Core Curriculum and Minimum Academic Standard (NUC CCMAS) for undergraduate Programme in Nigerian Universities (2024). Special considerations were given to the naming and content of the courses while at the same time accommodating the creation of new courses in line with the current evolving areas of the discipline globally.

Bearing in mind the importance of security in all aspect of computing, more optional security courses from sister Department of Cyber Security Science were added in addition to the existing compulsory Computer and Network security course. Also, considering the fact that we are in the era of data and digital economy; with the need to make sense of the data being generated on a daily basis, the Data Mining course has been revamped and now offered at 400 level of the programme. The curriculum has **70% Global Course Structure and 30% Additional Core Departmental Courses** with **FUTM-CPT** codes were introduced to provide solutions to local problems being encountered in computing discipline.

This new 2024-2028 curriculum edition maintains the total instructional hours and the total Units required for graduation as obtained in the previous editions. Therefore, the new curriculum does not overburden the students but seeks to make them well grounded in all the areas of the discipline.

E.F Aminu (PhD, MCPN, MNCS)
Head of Department

VISITOR AND PRINCIPAL OFFICERS OF THE UNIVERSITY

Visitor	His Excellency, Chief Bola Ahmed Adekunle Tinubu, GCFR President and Commander in-Chief of Armed Forces, Federal Republic of Nigeria.
Chancellor	His Royal Majesty Oba Aladetoyinbo Ogunlade Aladelusi
Pro-Chancellor & **Chairman, Governing Council**	Dr. Mohammed Kudu Santuraki DVM (ABU); MBA (UI), FCIoD, FNIM, FICA
Vice Chancellor	Professor Faruk Adamu Kuta B.Sc (UDUS);M.Tech (FUTMINNA); Ph.D.(ATBU)
Deputy Vice Chancellor (Academic)	Engr Professor Abdullahi Mohammed, FNIP, FICA B.Eng (ABU); M.Eng(FUTMINNA), PhD(UNITEN)
Deputy Vice Chancellor (Administration)	Professor Uno Essang Uno, FNIP, FICA B.Sc. (UNICAL); M.Sc., PhD(UNIABUJA)
Registrar	Mr Danladi Mallam B.A (Ed), M.A (UDUS) , MANUPA, NMIN
Bursar	Dr. (Mrs). Yunana Hadiza Goje *B.Sc., M.Sc (ABU), PhD. FCNA,FCTI*
University Librarian	Prof. Saka Katamba NCE, BLIS(ABU); MLS. (BUK); Ph.D(UNIMAID), CLN

PRINCIPAL OFFICERS OF THE SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY (SICT)

Prof. Jude Kur	-	Ag. Dean
Dr. Sulaimon Adebayo Bashir	-	Deputy Dean
Mrs. Grace Kolo	-	School Secretary

PAST AND PRESENT HEADS OF DEPARTMENT

1. Dr. (Mrs) Francisca N. Ogwueleka	-	2010 – 2012
2. Dr. Muhammad Bashir Abdullahi	-	2012 - 2019
3. Dr. Sulaimon A. Bashir	-	2019 - 2021
4. Dr. (Mrs.) Opeyemi Aderiike Abisoye	-	2021- 2025
5. Dr. Enesi Femi Aminu	-	2025- till date

TABLE OF CONTENTS

Title Page	1
Foreword	2
Principal Officers of the University	3
Principal Officers of the School	3
Past and Present Heads of Department	3
Table of Contents	4
CHAPTER ONE	
Background of Federal University of Technology, Minna	6
History of Department of Computer Science	6
Philosophy of the Department	6
Vision of the Department	7
Mission of the Department	7
Objectives of the Department	7
Competencies of the Graduates	7
Organizational Chart of the Department	8
CHAPTER TWO	
Admission Requirement	9
Duration of the Programme	9
Examination Regulations	9
Students' Welfare	10
Lateness to Classes	10
Mode of Examination	10
External Examination and Assessment	11
Conduct of Examinations	11
School Examination Officer	11
Carryover Courses	12
Dean's List, Probation and Withdrawal	12
Course Registration	12
Requirements for Graduation	13
Degree Awarded	13
Class of Degree	14
Weighting Courses	13
Grading System	13
Calculation of SGPA	14
Calculation of CGPA	14
Adding/Dropping of Courses	14
Intra and Inter University Transfers	14
Conditions for Progression	15
Regulations Guiding Professional Examinations	15
CHAPTER THREE	
Programme Structure	17
Course Content Specification	17

Course Synopsis 22
Distribution of Marks 41
Industrial Training 41

CHAPTER FOUR
Facilities Available 42
Staffing 42

CHAPTER ONE

BACKGROUND OF FEDERAL UNIVERSITY OF TECHNOLOGY, MINNA

Federal University of Technology, Minna, Niger State is a Federal Government owned university in Nigeria, established on 1st February,1983. The vision of the university is to become one of Nigeria's leading Universities and a Centre of excellence, recognized nationally, regionally and internationally for its quality, pedagogy, and research, which is supported by visionary leadership, responsible citizenship, internal and external partnership and a unique value system.

As a specialized University, its **mission** is committed to the training of skilled and innovative work force that would harness the forces of nature to transform Nigeria's natural resources into goods and services to be positively affect the economy and thus the quality of life of her people. The University's **objective** is to give effect to the nation's drive for the much-needed self-reliance in Science, Engineering and Technology. The University's **motto** is Technology for Empowerment. The current Vice-Chancellor is Professor Faruk Adamu Kuta

The University has two (2) Campuses: Bosso and Gidan-Kwano. The University runs a School system; an integrated unit of related disciplines with common academic interest in teaching and research. Presently, there are **Sixteen** (16) Schools namely:

- School of Agricultural Management and Extension Technology,
- School of Agronomy and Forestry Technology,
- School of Architectural Technology,
- School of Food Science and Agricultural Technology,
- School of Electrical Engineering and Technology (SEET),
- School of Environmental Technology (SET),
- School of Innovative Technology (SIT),
- School of Information and Communication Technology (SICT),
- School of Life Sciences (SLS),
- School of Physical Sciences (SPS),
- School of Science and Technology Education (SSTE),
- School of Basic Medical Sciences (SBMS),
- School of Allied Health Sciences (SAHS),
- School of Pharmaceutical Sciences (USPHS),
- School of Infrastructure, Process Engineering and Technology (SIPET)
- Postgraduate School (PGS).

A Dean and an Administrative head known as School Secretary, who works with the Dean and coordinates any administrative matter of the School, head each School.

DEPARTMENT OF COMPUTER SCIENCE

The Department of Computer Science was established in 2009. It was created from the defunct Department of Mathematics/Computer Science with the aim that it is the backbone of Information and Communication Technology (ICT). The Department offers a degree programme of Bachelor of Technology (B.Tech.) in Computer Science. The Department of Computer Science is one of the departments in the School of Information and Communication Technology (SICT). The

Department offers a unique educational opportunity for students to achieve excellence through vigorous classes, practical and participation in cutting edge ICT research.

BTech. Computer Science

Overview

The BTech. Computer Science programme teaches the essential skills of Computer Science emphasizing the core elements of Computer programming, System Analysis and Design, Networking, Software Development, Data Science and Futuristic Technology, demystifying and bringing patterns to life with practicals. Students of this programme are equipped with the study of the algorithmic process and the computational machines ranging from algorithms, practical issues in implementing computing systems in hardware as well as software. The graduates of this programme will understand the impact of computing and its applications.

Philosophy

The philosophy of Computer Science programme is to provide broad and high quality education that emphasizes the theoretical and algorithmic foundations of computing, which guide design, implementation and application of computation systems.

Objectives

The specific objectives are to:

i. create in students the awareness of and enthusiasm for Computer Science and its capabilities;

ii. provide students with a broad and balanced foundation of Computer Science knowledge and practical skills;

iii. prepare students to formulate real world problems in Computer Science, employ problem-solving skills and use appropriate tools and technologies to obtain valid and realistic solutions;

iv. develop in students the ability to analyze, evaluate and propose alternative solutions to given software and/or algorithm designs;

v. develop students' abilities in self-management and teamwork;

vi. prepare students to be proficient, professional and ethical in their careers;

vii. prepare students to communicate effectively both orally and in writing; and

viii. develop in students the ability to engage in life-long learning and growth in Computer Science and to be potential job creators.

Unique Features of the Programme

The unique features of the programme are:

i. deliberate emphasis on coverage and developing competence on the usage of opensource

software;

ii. additional hands-on practical component in a number of courses to emphasise students' engagement in the learning process for better learning and developmentof soft skills; and

iii. emphasis on formal methods and algorithmic coverage of computing concepts and principles.

iv. Employability Skills

In Nigeria, like in many other countries, there is an abundance of opportunities for people with computing skills. However, given the intense competition in the job market, a good Computer Science degree may be necessary but not sufficient for employment. In addition toa good degree, employers are increasingly requiring candidates to demonstrate employability skills such as communication and teamwork, organisation and management, critical thinking, leadership, technology skills and self-management. The courses in this programme have beentailored to help develop and enhance acquisition of these skills by graduates of the programme.

21st Century Skills

Among the 21st Century skills for the programme are:

i. creativity;

ii. information literacy;

iii. media literacy;

iv. flexibility;

v. social skills;

vi. Problem solving;

vii. collaboration;

viii. global awareness;

ix. innovation skills; and

x. critical thinking.

COMPETENCES OF THE GRADUATES

At the end of a successful completion of the B.Tech. Computer Science programme, our graduates should have the following abilities:

i. An ability to analyse a problem, identify and define the computing requirements appropriate to its solution.
ii. An ability to analyse the impact of computing on individuals, organizations, and society.
iii. An ability to design, implement and evaluate a computer-based system, process, component, or program to meet desired needs.
iv. An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modelling and design of computer-based systems in a way that demonstrates comprehension of the trade-offs in design choices.
v. An ability to live usefully with vision and entrepreneurial skills in this ICT age.

ORGANIZATIONAL CHART OF THE DEPARTMENT

The Department is headed by the Head of Department (HOD) who assigns duty and responsibilities to staff. The Departmental activities are guided by the regulations approved by the University Senate and supervised by the Dean of School. The organisation structure is shown in Figure 1. The department uses committee system with departmental board as its highest decision-making body after the committees. The HOD chairs all the meetings of the Departmental Board while the Secretary who is appointed from the members of academic staff takes the minutes, prepares and circulates it in readiness for the next meeting. Another staff assists him/her. The department staff are involved in any decision concerning issues affecting the department through participation at departmental meetings where these issues are deliberated on and suggestions made. The meeting considers and approves course allocations to academic staff, examination schedule and procedures, examination results, research areas and other matters concerning the department.

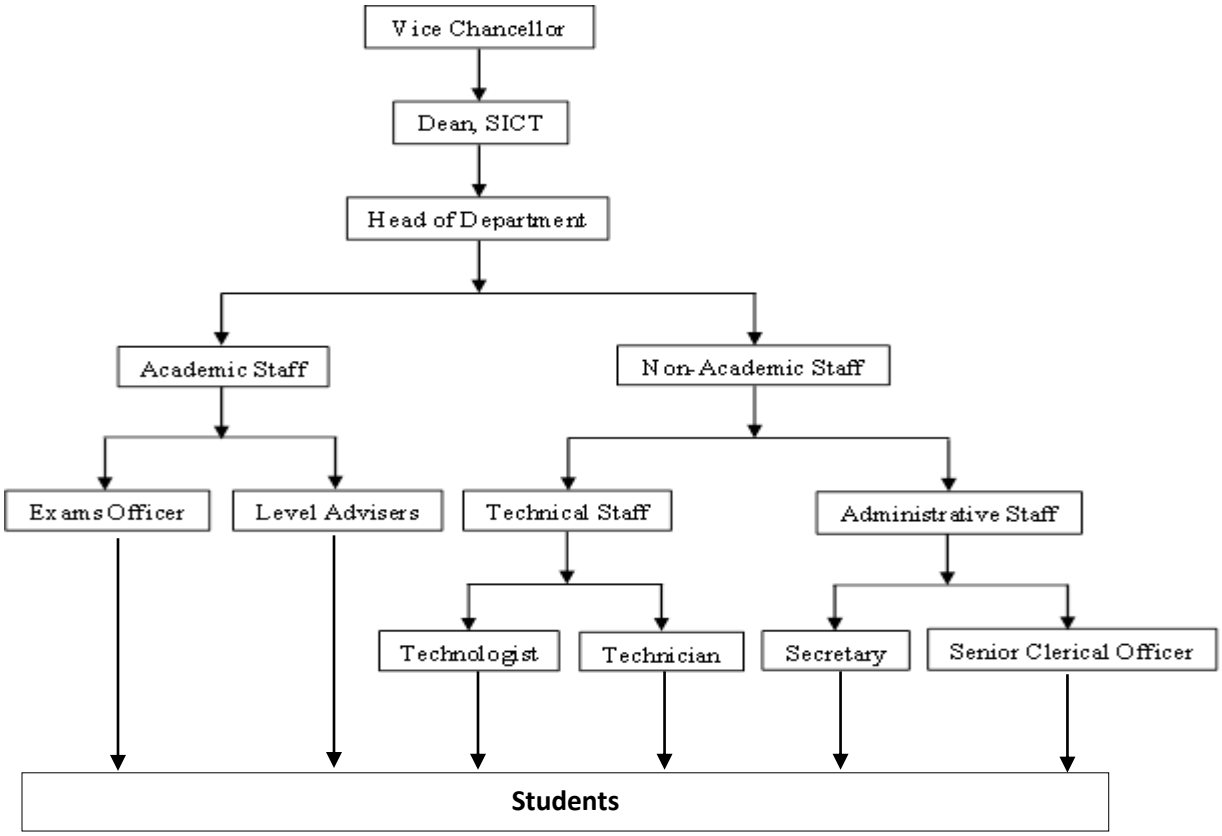

Figure 1: Organizational Chart of Computer Science

Some of the departmental committees are admission committee, research and seminar committee, finance committee, staff appraisal committee, curriculum review committee, welfare committee, strategic planning committee, etc.

CHAPTER TWO

ADMISSION AND GRADUATION REQUIREMENTS

4 Year Degree Programme

UTME

In addition to appropriate UTME-Score, a candidate must possess five Senior Secondary Certificate (SSC)-credits passes including English Language, Mathematics, Physics and any other relevant Science subjects in not more than two sittings.

3 Year Degree Programme

Direct Entry

A minimum of a credit at the University/National Diploma or NCE with other five Senior School Certificate (SSC) credit passes in relevant Science subjects three of which must be in English Language, Mathematics, Physics.

Specific Federal University of Technology Minna Admission Requirement

UTME

Candidates are required to obtain five (5) O'Level credits in English Language, Mathematics, Physics, Chemistry, and either Biology or Geography in NECO, WAEC or NABTEB in not more than two (2) sittings. In addition, candidates must obtain a high aggregate score from the University Post-UTME Screening Exercise greater than or equal to the cut-off point as might be stipulated from time to time by the University.

Direct Entry (DE)

i. Holders of National Diploma (ND) in Computer Science/Engineering with a minimum of Lower Credit may qualify for admission into 200 Level.
ii. Holders of HSC/GCE-A Level/IJMB with credit passes in Mathematics, Physics and Chemistry also qualify for 200 Level.
iii. Holders of Higher National Diploma (HND) with at least Upper Credit in Computer Science may qualify for admission into 300 Level.

Minimum duration

The minimum duration of the Computer Science degree programme is four academic sessionsfor UTME. However, it is three academic sessions for candidates admitted to the 200 Level.

Specific Federal University of Technology Minna Minimum Duration

Duration of the Programme

The Bachelor of Technology degree in Computer Science is designed to last for a minimum and maximum durations as stated below:

Mode of entry	Minimum (semesters)	Maximum (semesters)

UTME	10	15
200 Level (D.E.)	8	12
300 Level(D.E.)	6	9

Graduation requirements

To be eligible for the award of the Bachelor degree in Computer Science, 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 course units for students admitted to 200 level; and

iii. Attain a minimum CGPA of 1.00.

iv. To graduate, a student must be found worthy in character throughout the period of his/her studentship and must accumulate the total units prescribed for the programme from Core, Faculty and General Studies courses as well as SIWES, Seminar and Final Year Project.

Specific Federal University of Technology Minna Graduation Requirement

To be eligible for the award of the Bachelor degree in Computer Science, a student must have: passed all the core courses, university and faculty/school required courses andelectives; accumulated a minimum of 155 course units for students admitted through UTME, 121course units for students admitted to 200 level; 87 course units for students admitted to 300 level; and attain a minimum CGPA of 1.00.

To graduate, a student must be found worthy in character throughout the period of his/her studentship and must accumulate the total units prescribed for the programme from Core, Faculty and General Studies courses as well as SIWES, Seminar and Final Year Project.

MODE OF ENTRY	UTME	200 LEVEL (D. E.)	300 LEVEL (D. E.)
Basic Science	19	4	0
Core ICT courses	118	103	77
General Studies	12	8	4
SIWES	6	6	6
Total	**155**	**121**	**87**

EXAMINATION REGULATIONS

The Senate of the Federal University of Technology, Minna, approves the following Code of Conduct. Members of the Academic staff and student are expected to abide by the rules and regulations enunciated in the code of conduct. Only the Senate of the University shall amend the Code from time to time as the need arises. Any exception to the regulations shall be accepted only under extenuating circumstances and must be sanctioned by the Senate.

ATTENDANCE

1. Attendance in lectures, practical classes, fieldwork and SIWES shall be regarded as part of the requirements for a successful completion of a course. A student shall record a 75 percent attendance in lectures and 100 percent participation in practical, laboratory exercises, class assignments, SIWES and fieldwork in order to qualify to sit a semester's examination.
2. Therefore, each lecturer shall be required to keep attendance for each course. The student shall be made aware of this and shall be informed on an on-going basis during the course how he or she is doing in the course with respect to attendance. The mode employed in keeping the attendance shall be the prerogative of each lecturer and /or the respective departments.
3. Exemption from the above rules on attendance shall be on medical ground and shall be granted on submission of a certificate from the Director of the University Health Services. For reasons other than medical, Heads of Departments concerned in consultation with their teaching staff and ratified by the Dean must endorse student applications. Any student who fails to meet the above requirements in anyone course shall carryover that course.

STUDENT'S WELFARE

(a) Handling of academic grievances

The Students' academic grievances are normally channelled through the class representative and the respective level adviser to the Head of Department. The issue will then be discussed at the Departmental meeting, and where necessary, an ad-hoc committee may be constituted to look into the grievances. Issues that cannot be resolved at the Departmental level may be referred to the School Academic Board, where, with the permission of members, the Dean might set up a committee to look into the issue.

(b) Students' academic advising

The Head of Department appoints one academic staff as an adviser to each level. Moreover, there is an intimate interaction between the students and the staff of the Department through the students' advisers. Consequently, the students have direct access to the staff and the Head of Department always ready to come to the assistance of the students, where and when necessary.

LATENESS TO CLASS

Lecturers are expected to start on the exact time they are scheduled. Any student who is not in the class 15 minutes after the lecturer has started his/her lecture shall be deemed to be absent from that class. Also, the students shall wait for the lecturer for up to 15minutes; starting from the time the lecture is scheduled to commence. If a lecturer fails to turn up for lectures or is fond of lateness in attending to his lectures the class representative should intimate the HOD Computer Science or the Dean SICT.

MODE OF EXAMINATION

The following three types of assessment may be used to examine a course in Computer Science:

1. Practical examination
2. Oral examination
3. Written examination/E-examination

The particular mode of examination or combinations chosen shall be dictated by the type of course and shall be the prerogative of the lecturer and the department.

EXTERNAL EXAMINATION AND ASSESSMENT

The examination questions are set and marked by the Course lecturer(s). The Department internally moderates the questions. Course lecturer(s) evaluate(s) and grade(s) the students' answers scripts using marking schemes and model answers, while the Head of Department carries out double-checking of the results submitted by the course lecturers. On request, the Department issues students their results after examinations having been approved by the Senate.

In the case of the final year degree examinations, the questions with solutions shall be sent out to the University appointed External Examiner for external moderation. Qualified External examiners (the proposed and the alternate) shall be nominated by the Head of Department and submitted to the University Senate for approval through the SICT school board. The approved External examiner shall be appointed for one academic session and may be reappointed for a maximum of three terms. Only when qualified and appointable external examiner is not available from within Nigeria shall such appointment be extended to examiners from outside the Country.

CONDUCT OF EXAMINATIONS

1. Examination shall be conducted by the school. Invigilators shall be chosen by the School Examination Officer from among the academic and senior technical members of staff of the various Departments. For each examination, there is a Chief Invigilator, who is a senior academic staff among the invigilators. There shall be at least two invigilators for each examination and for examination involving more than 50 candidates, one extra invigilator shall be required for each additional 25 candidates. No lecturer shall be allowed to be the Chief Invigilator for his course.
2. The functions of the various officers charged with the responsibility of conducting the examinations are defined as follows:

SCHOOL EXAMINATION OFFICER

The School Examination Officer shall not below the rank of Lecturer I, and shall be appointed by the Senate on the recommendation of the School (SICT) Board. It is his/her responsibility to:

(i) Prepare the Examination timetable and assign Chief invigilator/invigilators to each examination.

(ii) Prepare attendance slips, attendance registers and receipts for all examinations

(iii) Notify the Academic Officer in advance of all the materials that would be needed for each semester examination, e.g. answer booklets, answer sheets, graph papers etc. and to collect all these materials one week to the commencement of the semester examination.

(iv) Handover the envelope(s) containing the question papers to the Chief invigilator about 30 minutes before the commencement of any examination scheduled for the period.

(v) Collect all the answer scripts, the attendance slips and attendance register from the Chief invigilator after making sure that the total number of answer scripts tally with the number of students present at the end of the examination. A receipt must be issued for all items collected from him/her.

(vi) Handover the answer scripts to the Departmental Examination Officer concerned with the examination after he/she has signed for them.

CARRYOVER COURSE

No student is allowed to carry over any course in which he/she scored an “E” grade or above. Both the old and new grades in a carryover course shall be retained in the student’s transcript and they will also be used in computing his/her CGPA.

DEAN'S LIST, PROBATION AND WITHDRAWAL

1. Within any one semester, any student that registers a CGPA of 4.0 and above shall qualify to be on the Dean's list of exceptional students. A letter shall be issued from the Dean's Office to inform the student.
2. If a student recorded a CGPA below 2.0 at the end of 100Level, he/she shall be withdrawn from the Department. While if such a student is in 200level and above; he/she shall be placed on probation at the first instance (SP1) and if recorded consecutively for the second time; he/she shall be withdrawn from the University.

COURSE REGISTRATION

The level adviser for each level is also the registration officer for that level. Every course is assigned a Unit load that corresponds with the number of lecture hours per week required to complete the course during the semester. A course that requires two hours of lectures per week shall be assigned two Unit units and vice versa.

Students are expected to register the courses they are to offer in a particular semester/session. Carry over courses are registered first before the fresh courses for that semester.

The maximum unit a student is allowed to register for in a semester is 18 units and a minimum of 15 Unit units would be allowed in any semester. This means that the total number of Unit units for all courses registered for by any student during a semester will not exceed 18 or be below 15 units.

DEGREE AWARDED

The nomenclature of the first degree awarded by the Department is Bachelor of Technology (Computer Science).

CLASS OF DEGREE

The class of degree obtained at the end of the undergraduate programme is classified as shown in Table 1:

Table 1: Class of degree

CLASS OF DEGREE	CGPA
First Class	4.50 – 5.00
2nd Class Upper	3.50 – 4.49
2nd Class Lower	2.40 – 3.49
Third Class	1.50 – 2.39
Pass	1.00 – 1.49
Fail	0.00 – 0.99

WEIGHTING COURSES

The assessment of a student's performance in all courses shall be as follows:

(a) Continuous Assessment = 40%
(b) Examination = 60%
100%

GRADING SYSTEM

The Federal University of Technology, Minna operates a 5-point grading system. Table 2 shows the letter grades as in use under the grading system.

Table 2: Grading system

LETTER	GRADE	SCORE (MARKS)	GRADE POINTS
A	Excellent	70 – 100	5
B	Very Good	60 – 69	4
C	Good	50 – 59	3
D	Intermediate	45 – 49	2
E	Fair	40 – 44	1
F	Failure	0 – 39	0

CALCULATION OF SEMESTERIAL GRADE POINT AVERAGE (SGPA)

Table 3: Student's First Semester Results

COURSE	GRADE OBTAINED	UNIT UNIT (X1)	POINT OBTAINED (Y1)	GRADE POINT (X1*Y1)
MTH101	B	3	4	12
PHY101	A	2	5	10
GST111	D	2	2	2
FUTM-CPT111	C	3	3	9
TOTAL		**10**		**33**

$$SGP = \sum (X1 * Y1) = 33$$

$$SGPA = \frac{SGP}{\sum X1} = \frac{33}{10} = 3.3$$

CALCULATION OF CUMULATIVE GRADE POINT AVERAGE (CGPA)

Table 4: Student's Second Semester Results

COURSE	GRADE OBTAINED	UNIT UNITS (X2)	POINT OBTAINED (Y2)	GRADE POINT (X2*Y2)
COS102	A	3	5	15
MTH102	C	3	3	9
PHY102	B	2	4	8
FUTM-CPT122	E	3	1	3
TOTAL		**11**		**35**

$$CGP = \sum \left(SGP\left(\text{First semester}\right) + SGP\left(\text{Second semester}\right) \right) = 33 + 35 = 68$$

$$CGPA = \frac{CGP}{\sum X1 + \sum X2} = \frac{68}{10+11} = \frac{68}{21} = 3.24$$

ADDING/DROPPING OF COURSES

Since a student is required to attain a minimum of 75% class attendance in order to qualify for the semester examination, subject/course combinations can only be changed within the first four weeks of the semester in the first year as the case may be, in line with the University current regulations.

INTRA AND INTER UNIVERSITY TRANSFERS

All candidates seeking transfer (whether intra or inter University) must have spent a minimum of one academic session in the programme of first admission with full Sessional result attached to the application for transfer.

(a) All intra University candidates seeking transfers to the Department of Computer Science from other Departments, must

  (i) Have a minimum CGPA of 3.00.

  (ii) Not have more than six (6) units deficiencies in the previous courses taken.

  (iii) Not scored less than a C grade in CPT111 (Introduction to Computer Science) and CPT121 (Introduction to Programming).

  (iv) Have fulfilled O'Level requirements stipulated for the UTME candidates.

(b) All inter University candidates seeking transfers to the Department of Computer Science of the Federal University of Technology, Minna must:

  (i) Have fulfilled O'Level requirements stipulated for UTME candidates.
  (ii) Be studying an Engineering/Science programme in their current university.
  (iii) Have passed all courses registered in their current university before seeking the transfer.
  (iv) Have a minimum CGPA of 4.00 out of 5.00 or 3.00 on a scale of 4.00.
  (v) Transfer cases can only be entertained up to and not beyond 200L.

Notice: In all cases, admission is purely based on:

  (a) Available vacancies.

  (b) The number of candidates applying for admission.

CONDITIONS FOR PROGRESSION

Continuing students: The case of screening and weeding is not limited to new students only. The following criteria were approved for the continuing students:

(a) Students crossing from 100L to 200L must have passed 6 out of 8 unit loads for each of Mathematics and Physics and 3 out of 6 unit loads of Chemistry.

(b) For a student to proceed to 200L he/she must have a CGPA of not less than 1.00. Else, he/she will be asked to withdraw from the department.

(c) A student is not allowed to carry over more than 16 units or eight (8) courses per session otherwise; such a student would be advised to repeat the session.

(d) A student with a CGPA of less than 1.00 for a semester or a session in the first instance; is placed on semesterial probation (P) or sessional probation (SP1), respectively. Consequently, if the student is still on probation for two consecutive sessions, he/she will be asked to withdraw from the university.

REGULATIONS GUIDING PROFESSIONAL EXAMINATIONS

In line with the Computer Professionals (Registration Council of Nigeria) (CPN) regulation and the Nigerian Computer Society (NCS) all graduates of the Department of Computer Science are statutorily obligated to register with the Council and/or the Society before practicing.

CHAPTER THREE

PROGRAMME STRUCTURE

The program structure for Computer Science from 100 Level to 500 Level is as follows:

100 Level – Students are expected to take relevant courses in basic science such as Mathematics, Physics, Chemistry, and General Studies, which will serve as foundation upon which the courses at subsequent levels are to be based. In addition, introductory courses in Computer, Environmental Science and Biology are included.

200 Level – Students at this level take Computer Science courses as well as courses in Mathematics, Statistics and Physics.

300 Level – Students at this level take more Computer Science courses with a General Studies course.

400 Level – Students at this level take Detailed and specific courses in Computer Science together with practical. In the second semester, the students undertake Industrial Work Experience Scheme (SIWES) for six months.

500 Level – At this level, the students take more detailed and specific courses in Computer Science, with seminar presentation and research project. The students are equally passed through oral defense and examination by the Department's external examiner.

COURSE CONTENT SPECIFICATION

Global Course Structure 70% and 30% Additional

LH= Lecture Hour **PH=** Practical Hour

100 Level

First Semester

Course Code	Course Title	Unit(s)	Status	LH	PH
GST111	Communication in English	2	C	15	45
MTH101	Elementary Mathematics I	3	C	45	0
PHY101	General Physics I	2	C	30	0
PHY107	General Practical Physics I	1	C	0	45
COS101	Introduction to Computing Sciences	3	C	30	45

FUTM-CPT111	Probability for Computer Science	3	C	45	0
FUTM-CPT112	Front-End Web Development	3	C	30	45
	Total	**17**			

100 Level

Second Semester

Course Code	Course Title	Unit(s)	Status	LH	PH
GST112	Nigerian Peoples & Culture	2	C	30	0
MTH102	Elementary Mathematics II	3	C	45	0
PHY102	General Physics II	2	C	30	0
PHY108	General Practical Physics II	1	C	0	45
STA 111	Descriptive Statistics	3	C	45	0
COS102	Introduction to Problem Solving	3	C	30	45
FUTM-CPT122	Introduction to Computer Hardware	3	C	30	45
	Total	**17**			

200 Level

First Semester

Course Code	Course Title	Unit(s)	Status	LH	PH
GST 212	Philosophy, Logic and Human Existence	2	C	30	0
ENT 211	Entrepreneurship and Innovation	2	C	30	0
MTH 201	Mathematical Methods I	2	C	30	0
COS 201	Computer Programming I	3	C	30	45
IFT 211	Digital Logic Design	2	C	15	45
SEN 201	Introduction to Software Engineering	2	C	30	0
FUTM-CPT 211	Back -End Web Development	2	C	15	45
FUTM-CPT 212	Inferential Statistics for Computer Science	2	C	15	45
Total		**17**			

200 Level

Second Semester

Course Code	Course Title	Unit(s)	Status	LH	PH
MTH 202	Elementary Differential Equations	2	C	30	0
COS 202	Computer Programming II	3	C	30	45
CSC 203	Discrete Structures	2	C	30	0
IFT 212	Computer Architecture and Organization	2	C	15	45
FUTM-CPT 221	Numerical Computation for Computer Science	2	C	15	45
FUTM-CPT 222	Server-Side Web Development	3	C	30	45
FUTM-CPT 223	System Concept and Design	3	C	30	45
Total		**17**			

300 Level

First Semester

Course Code	Course Title	Unit(s)	Status	LH	PH
CSC 308	Operating Systems	3	C	30	45
CSC 301	Data Structures	3	C	30	45
CSC 309	Artificial Intelligence	2	C	15	45
CYB 201	Introduction to Cyber security and Strategy	2	C	30	0
ICT 305	Data Communication System & Network	3	C	30	45
FUTM-CPT 311	Programming Language Translation and Compiler Design	3	C	45	0
FUTM-CPT 312	Object-Oriented Analysis, Design and Implementation.	2	C	15	45
Total		**18**			

300 Level

Second Semester

Course Code	Course Title	Unit(s)	Status	LH	PH
GST 312	Peace and Conflict Resolution	2	C	30	0
ENT 312	Venture Creation	2	C	15	45
CSC 322	Computer Science Innovation and New Technologies	2	C	15	45
DTS 304	Data Management I	3	C	30	45
FUTM-CPT 321	Human Computer Interaction	3	C	30	45
FUTM-CPT 322	Theory of Computing	3	C	30	45
FUTM-CPT 323	Mobile Application Development for Android Platform	3	C	30	45
Total		**18**			

400 Level

First Semester

Course Code	Course Title	Unit(s)	Status	LH	PH
COS 409	Research Methodology and Technical Report Writing	3	C	45	0
CSC 401	Algorithms and Complexity Analysis	2	C	30	0
CSC 402	Ethics and Legal Issues in Computer Science	2	C	30	0
INS 401	Project Management	2	C	30	0
FUTM-CPT 411	Introduction to Machine Learning for Data Mining	3	C	30	45
FUTM-CPT 412	Advanced Visual Programming with VB.net	3	C	30	45
Total		**15**			

400 Level

Second Semester

Course Code	Course Title	Unit(s)	Status	LH	PH
CSC 499	SIWES I and II	6	C		
Total		**6**			

500 Level

First Semester

Course Code	Course Title	Unit(s)	Status	LH	PH
FUTM-CPT 511	Introduction to Cloud Computing	3	C	30	45
FUTM-CPT 512	Operation Research	3	C	45	0
FUTM-CPT 513	Introduction to Applied Neural Network and Deep Learning	3	C	30	45
FUTM-CPT 514	Introduction to Natural Language Processing	3	C	30	45
FUTM-CPT 515	Advanced Database Systems	3	C	30	45
	Total	**15**			

500 Level

Second Semester

Course Code	Course Title	Unit(s)	Status	LH	PH
FUTM-CPT 521	Big Data Analytics	3	C	30	45
FUTM-CPT 522	Computer System Performance Evaluation	3	C	45	0
FUTM-CPT 523	Introduction to Computer Vision	3	C	30	45
CSC 598	Final Year Project I and II	6	C		
	Total	**15**			

Course Contents and Learning Outcomes

100 Level

GST 111: Communication in English (2 Units, C: LH-15; PH-45)

Learning Outcomes

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

i. identify possible sound patterns in English language;
ii. list notable language skills;
iii. classify word formation processes;
iv. construct simple and fairly complex sentences in English;
v. apply logical and critical reasoning skills for meaningful presentations;
vi. demonstrate an appreciable level of the art of public speaking and listening; and
vii. write simple and technical reports.

Course Contents

Sound patterns in English Language (vowels and consonants, phonetics and phonology). English word classes (lexical and grammatical words, definitions, forms, functions, usages, collocations). Sentence in English (types: structural and functional, simple and complex). Grammar and Usage (tense, mood, modality and concord, aspects of language use in everydaylife). Logical and Critical Thinking and Reasoning Methods (Logic and Syllogism, Inductive andDeductive Argument and Reasoning Methods, Analogy, Generalisation and Explanations). Ethical considerations, Copyright Rules and Infringements. Writing Activities: (Pre-writing, writing, post writing, editing and proofreading; brainstorming, outlining, paragraphing. Typesof writing, Summary, Essays, Letter, Curriculum Vitae, Report writing, Note making, etc. Mechanics of writing). Comprehension Strategies: (Reading and types of Reading, Comprehension Skills, 3RsQ). Information and Communication Technology in modern language learning. Language skills for effective communication. Major word formation processes. Writing and reading comprehension strategies. Logical and critical reasoning for meaningful presentations. Art of public speaking and listening. Report writing.

GST 112: Nigerian Peoples and Culture (2 Units, C: LH-30)

Learning Outcomes

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

i. analyse the historical foundation of the Nigerian culture and arts in pre-colonial times;

ii. list and identify the major linguistic groups in Nigeria;

iii. explain the gradual evolution of Nigeria as a political unit;

iv. analyse the concepts of Trade, Economic and Self-reliance status of the Nigerian peoples towards national development;

v. enumerate the challenges of the Nigerian State towards Nation building;

vi. analyse the role of the Judiciary in upholding people's fundamental rights;

vii. identify acceptable norms and values of the major ethnic groups in Nigeria; and

viii. list and suggest possible solutions to identifiable Nigerian environmental, moral and value problems.

Course Contents

Nigerian history, culture and art up to 1800 (Yoruba, Hausa and Igbo peoples and culture; peoples and culture of the ethnic minority groups). Nigeria under colonial rule (advent of colonial rule in Nigeria; Colonial administration of Nigeria). Evolution of Nigeria as a political unit (amalgamation of Nigeria in 1914; formation of political parties in Nigeria; Nationalist movement and struggle for independence). Nigeria and challenges of nation-building (militaryintervention in Nigerian politics; Nigerian Civil War). Concept of trade and economics of self- reliance (indigenous trade and market system; indigenous apprenticeship system among Nigeria people; trade, skill acquisition and self-reliance). Social justice and national development (law definition and classification). Judiciary and fundamental rights. Individual, norms and values (basic Nigeria norms and values, patterns of citizenship acquisition; citizenship and civic responsibilities; indigenous languages, usage and development; negative attitudes and conducts. Cultism, kidnapping and other related social vices). Re-orientation, moral and national values (The 3R's – Reconstruction, Rehabilitation and Re-orientation) Re-orientation Strategies: Operation Feed the Nation (OFN), Green Revolution, Austerity Measures, War Against Indiscipline (WAI), War Against Indiscipline and Corruption (WAIC), Mass Mobilisation for Self-Reliance, Social Justice and Economic Recovery (MAMSER), NationalOrientation Agency (NOA). Current socio-political and cultural developments in Nigeria.

MTH 101: **Elementary Mathematics I (Algebra and Trigonometry**) (2 Units, C: LH-30)

Learning Outcomes

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

i. explain basic definition of Set, Subset, Union, Intersection, Complements and use of Venn diagrams;

ii. solve quadratic equations;

iii. solve trigonometric functions;

iv. identify the various types of numbers; and

v. solve some problems using Binomial theorem.

Course Contents

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

MTH 102: Elementary Mathematics II (Calculus) (2 Units, C: LH-30)

Learning Outcomes

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

i. distinguish types of rules in Differentiation and Integration;
ii. describe the meaning of Function of a real variable, graphs, limits and continuity; and
iii. solve some applications of definite integrals in areas and volumes.

Course Contents

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.

PHY 101: General Physics I (Mechanics) (2 Units C: LH-30)

Learning Outcomes

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

i. identify and deduce the physical quantities and their units;
ii. differentiate between vectors and scalars'
iii. describe and evaluate motion of systems on the basis of the fundamental laws of mechanics;
iv. apply Newton's laws to describe and solve simple problems of motion;
v. evaluate work, energy, velocity, momentum, acceleration, and torque of moving or rotating objects;
vi. explain and apply the principles of conservation of energy, linear and angular momentum;
vii. describe the laws governing motion under gravity; and
viii. explain motion under gravity and quantitatively determine behaviour of objects moving under gravity.

Course Contents

Space and time. Units and dimension, Vectors and Scalars, Differentiation of vectors. Displacement, velocity and acceleration. Kinematics. Newton laws of motion (Inertial frames,

Impulse, force and action at a distance, momentum conservation). Relative motion. Application of Newtonian mechanics. Equations of motion. Conservation principles in physics,Conservative forces, conservation of linear momentum, Kinetic energy and work, Potential energy, System of particles, Centre of mass. Rotational motion. Torque, vector product, moment, rotation of coordinate axes and angular momentum. Polar coordinates. Conservationof angular momentum. Circular motion. Moments of inertia, gyroscopes and precession. Gravitation: Newton's Law of Gravitation, Kepler's laws of planetary motion, Gravitational potential energy, Escape velocity, Satellites motion and orbits.

PHY 102: General physics II (Electricity & magnetism) (2 Units, C: LH-30)

Learning Outcomes

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

i. describe the electric field and potential, and related concepts, for stationary charges;
ii. calculate electrostatic properties of simple charge distributions using Coulomb's law, Gauss's law, and electric potential;
iii. describe and determine the magnetic field for steady and moving charges;
iv. determine the magnetic properties of simple current distributions using Biot-Savart and
v. Ampere's law;
vi. describe electromagnetic induction and related concepts and make calculations using
vii. Faraday and Lenz's laws;
viii. explain the basic physical of Maxwell's equations in integral form;
ix. evaluate DC circuits to determine the electrical parameters;
x. determine the characteristics of ac voltages and currents in resistors, capacitors, and Inductors.

Course Contents

Forces in nature. Electrostatics (electric charge and its properties, methods of charging). Coulomb's law and superposition. Electric field and potential. Gauss's law. Capacitance. Electric dipoles. Energy in electric fields. Conductors and insulators. DC circuits (current, voltage and resistance. Ohm's law. Resistor combinations. Analysis of DC circuits. Magnetic fields. Lorentz force. Biot-Savart and Ampère's laws. Magnetic dipoles. Dielectrics. Energy in magnetic fields. Electromotive force. Electromagnetic induction. Self and mutual inductances.Faraday and Lenz's laws. Step up and step down transformers. Maxwell's equations. Electromagnetic oscillations and waves. AC voltages and currents applied to inductors, capacitors, and resistance.

PHY 107: General Practical Physics I (1 Unit, C: PH-45)

Learning Outcomes

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

i. conduct measurements of some physical quantities;
ii. make observations of events, collect and tabulate data;
iii. identify and evaluate some common experimental errors;
iv. plot and analyse graphs; and
v. draw conclusions from numerical and graphical analysis of data.

Course Contents

This introductory course emphasizes quantitative measurements, the treatment of measurement errors and graphical analysis. A variety of experimental techniques should be employed. The experiments include studies of meters, the oscilloscope, mechanical systems, electrical and mechanical resonant systems, light, heat, viscosity etc., covered in PHY 101 and PHY 102. However, emphasis should be placed on the basic physical techniques for observation, measurements, data collection, analysis and deduction.

PHY 108: General Practical Physics II (1 Unit, C: PH-45)

Learning Outcomes

On completion, the student should be able to:

i. conduct measurements of some physical quantities;
ii. make observations of events, collect and tabulate data;
iii. identify and evaluate some common experimental errors;
iv. plot and analyse graphs;
v. draw conclusions from numerical and graphical analysis of data; and
vi. prepare and present practical reports.

Course Contents

This practical course is a continuation of PHY 107 and is intended to be taught during the second semester of the 100 level to cover the practical aspect of the theoretical courses thathave been covered with emphasis on quantitative measurements, the treatment of measurement errors, and graphical analysis. However, emphasis should be placed on the basic physical techniques for observation, measurements, data collection, analysis and deduction.

STA 111: Descriptive Statistics (3 Units, C: LH-45)

Learning Outcomes

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

i. explain the differences between permutation and combination;
ii. explain the concept of random variables and relate it to probability and distribution functions;
iii. describe the basic distribution functions; and
iv. explain the concept of exploratory data analysis.

Course Contents

Permutation and combination. Concepts and principles of probability. Random variables. Probability and distribution functions. Basic distributions: Binomial, geometric, Poisson, normal and sampling distributions; exploratory data analysis.

COS 101: Introduction to Computing Sciences (3 Units, C: LH-30; PH-45)

Learning Outcomes

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

i. explain basic components of computers and other computing devices;
ii. describe the various applications of computers;
iii. explain information processing and its roles in the society;
iv. describe the Internet, its various applications and its impact;
v. explain the different areas of the computing discipline and its specializations; and
vi. demonstrate practical skills on using computers and the internet.

Course Contents

Brief history of computing. Description of the basic components of a computer/computing device. Input/Output devices and peripherals. Hardware, software and human ware. Diverseand growing computer/digital applications. Information processing and its roles in society. The Internet, its applications and its impact on the world today. The different areas/programsof the computing discipline. The job specializations for computing professionals. The future ofcomputing.

Lab Work: Practical demonstration of the basic parts of a computer. Illustration of different operating systems of different computing devices including desktops, laptops, tablets, smart boards and smart phones. Demonstration of commonly used applications such as word processors, spreadsheets, presentation software and graphics. Illustration of input and outputdevices including printers, scanners, projectors and smartboards. Practical demonstration of the Internet and its various applications. Illustration of browsers and search engines. How toaccess online resources.

COS 102: Problem Solving (3 Units, C: LH-30; PH-45)

Learning Outcomes

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

i. explain problem solving processes;

ii. demonstrate problem solving skills;
iii. describe the concept of algorithms development and properties of algorithms;
iv. discuss the solution techniques of solving problem;
v. solve computer problems using algorithms, flowcharts, pseudocode; etc.; and
vi. solve problems using programming language using C, PYTHON, etc.

Course Contents

Introduction to the core concepts of computing. Problems and problem-solving. The identification of problems and types of problems (routine problems and non-routine problems). Method of solving computing problems (introduction to algorithms and heuristics).Solvable and unsolvable problems. Solution techniques of solving problems (abstraction, analogy, brainstorming, trial and error, hypothesis testing, reduction, literal thinking, means-end analysis, method of focal object, morphological analysis, research, root cause analysis, proof, divide and conquer). General Problem-solving process. Solution formulation and design:flowchart, pseudocode, decision table, decision tree. Implementation, evaluation and refinement. Programming in C, Python etc.

Lab Work: Use of simple tools for algorithms and flowcharts; writing pseudocode; writing assignment statements, input-output statements and condition statements; demonstrating simple programs using any programming language (Visual Basic, Python, C)

FUTM-CPT 111: Probability for Computer Science (3 Units; C; LH-45, PH-0)

Learning Outcome

On completion of this course, students should be able to:

i. count and arrange numbers using combinatorial techniques
ii. apply the knowledge of probability in solving practical problems
iii. differentiate between the likelihood and the actual outcome
iv. list a number of areas to which probability is applicable
v. state at least 2 applications of probability in machine learning and computer theorem.

Course Content

Counting and Combinatorics. Counting. permutation. combination. bucking/group assignment. Discrete Probability. Random Variables. discrete. continuous and multiple random variables. Distribution. discrete. normal. conditional and beta distributions. Point Estimation. Definition of a point estimator. properties of point estimator. Point estimator vs interval estimator. Point estimates methods. Limit Theorems. Chebyshev's theorem. the law of large numbers. the central limit theorem.

FUTM-CPT 121: Front-End Web Development (3 Units; C; LH-30, PH-45)

Learning Outcomes

On completion of the course, students should be able to:

i. outline the architecture of how world-wide-web works as a client-server application.
ii. state at least two roles of web servers and web browsers in web application.
iii. list and use at least 20 HTML tags for creating websites.
iv. list and apply at least 15 cascading style sheet style attributes to styling websites.
v. identify the use of Notepad, VSCode, Chrome, Mozilla Firefox etc. in developing and testing a website.
vi. debug and validate a webpage using w3c validator.
vii. itemize the steps for publishing a website to a webserver and how to update published website.

Course Contents.

Introduction to the Web: What is the Web and how does it works. The roles of web servers and web browsers. Tools needed for web development. Code editors (Notepad. Sublime Text. Dreamweaver. etc). Browsers (Chrome. Mozilla. Microsoft Edge. etc.). Dealing with files (html files. asset files. css files javascript files). Setting up different directory to store related files. Planning a website: website content. fonts and colours. images. etc. HTML for page structuring and defining semantics: What is HTML. Anatomy of an HTML element. nesting elements within other elements. block vs inline elements. Void elements. HTML document structure.. Head section. HTML text tags. Creating hyperlinks. HTML sematic tags for website structure. Nav. Aside. Footer. Section. Embedding images. Video and content. HTML tables. HTML forms. Introduction to CSS: What is CSS. Adding CSS to documents. Inline and external CSS. CSS structure . CSS selectors. Cascade specifity and inheritance. The box model styling text and fonts. styling lists. Styling links. Web fonts.CSS layoit. normal flow. Flexbox. Grids. Floats positioning. Multiple columns layout. Responsive design. Introduction to media queries. Legacy layout methods.

Practical: Extensive practical session will involve using code editors like VScode to write HTML code to design website. and styling the website with CSS.

FUTM-CPT 122: Introduction to Computer Hardware Systems and Maintenance (3 Units; C; LH-30, PH-45)

Learning Outcomes

On completion of the course, students should be able to:

i. state at least five parts of computer
ii. describe two types of computer maintenance
iii. state three kinds of computer viruses
iv. describe two computer programs

v. differentiate between troubleshooting and repair

vi. compare computer primary memory and secondary memory

Course Contents

Definitions and Computer Basics. Computer programs and their types. Introduction to different parts of computer system. Identification of Computer Parts. Introduction to software components of the Computer System. Assembling Computer System. Software Installation. Computer Maintenance. The components that require inspection during hardware maintenance procedures. Setting up or customizing a computer. Laptops maintenance. Virus and Malware Prevention and Removal. Computer troubleshooting and repair basics. Two categories of computer hardware preventative maintenance (System level maintenance and physical level maintenance). Corrective Maintenance. Primary Memory and Secondary Memory. Peripheral Hardware Use and Maintenance. Improving Slow Performance. Fixing Software and Hardware Problems. Troubleshooting and Repairing Printers.

200 Level

GST 212: Philosophy, Logic and Human Existence (2 Units, C: LH-30)

Learning Outcomes

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

i. know the basic features of philosophy as an academic discipline;

ii. identify the main branches of philosophy & the centrality of logic in philosophicaldiscourse;

iii. know the elementary rules of reasoning;

iv. distinguish between valid and invalid arguments;

v. think critically and assess arguments in texts, conversations and day-to-daydiscussions;

vi. critically asses the rationality or otherwise of human conduct under different existential conditions;

vii. develop the capacity to extrapolate and deploy expertise in logic to other areas of knowledge, and

viii. guide his or her actions, using the knowledge and expertise acquired in philosophyand logic.

Course Contents

Scope of philosophy; notions, meanings, branches and problems of philosophy. Logic as an indispensable tool of philosophy. Elements of syllogism, symbolic logic the first nine rules of inference. Informal fallacies, laws of thought, nature of arguments. Valid and invalid arguments, logic of form and logic of content deduction, induction and inferences. Creative and critical thinking. Impact of philosophy on human existence. Philosophy and politics, philosophy and

human conduct, philosophy and religion, philosophy and human values, philosophy and character molding, etc.

ENT 211: Entrepreneurship and Innovation (2 Units, C: LH-15; PH-45)

Learning Outcomes

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

i. explain the concepts and theories of entrepreneurship, intrapreneurship, opportunity seeking, new value creation, and risk taking;

ii. state the characteristics of an entrepreneur;

iii. analyse the importance of micro and small businesses in wealth creation, employment, and financial independence;

iv. engage in entrepreneurial thinking;

v. identify key elements in innovation;

vi. describe stages in enterprise formation, partnership and networking including business planning;

vii. describe contemporary entrepreneurial issues in Nigeria, Africa and the rest of the world; and

viii. state the basic principles of e-commerce.

Course Contents

Concept of Entrepreneurship (Entrepreneurship, Intrapreneurship/Corporate Entrepreneurship). Theories, Rationale and relevance of Entrepreneurship (Schumpeterian and other perspectives, risk-taking, necessity and opportunity-based entrepreneurship and creative destruction). Characteristics of Entrepreneurs (Opportunity seeker, risk taker, natural and nurtured, problem solver and change agent, innovator and creative thinker). Entrepreneurial thinking (Critical thinking, Reflective thinking, and Creative thinking). Innovation (Concept of innovation, Dimensions of innovation, Change and innovation, Knowledge and innovation). Enterprise formation, partnership and networking (Basics of business plan, Forms of business ownership, business registration and forming alliances and joint ventures). Contemporary Entrepreneurship Issues (knowledge, skills and technology, intellectual property, virtual office, networking). Entrepreneurship in Nigeria (Biography of inspirational entrepreneurs, youth and women entrepreneurship, Entrepreneurship support institutions, Youth enterprise networks and environmental and cultural barriers to entrepreneurship). Basic principles of e-commerce.

MTH 201: **Mathematical Methods I** (2 Units, C: LH-30)

Learning Outcomes

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

i. describe Real-valued functions of a real variable;

ii. solve some problems using Mean value Theorem and Taylor Series expansion; and

iii. evaluate Line Integral, Surface Integral and Volume Integrals.

Course Contents

Real-valued functions of a real variable. Review of differentiation and integration and their applications. Mean value theorem. Taylor series. Real-valued functions of two and three variables. Partial derivatives chain rule, extrema, Lagrangian multipliers. Increments, differentials and linear approximations. Evaluation of line, integrals. Multiple integrals.

MTH 202: **Elementary Differential Equations** (2 Units, C: LH-30)

Learning Outcomes

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

i. define the following: order and degree of a differential equation;

ii. describe some techniques for solving first and second order linear and non-linear equations; and

iii. solve some problems related to geometry and physics.

Course Contents

Derivation of differential equations from primitive, geometry, physics, etc. order and degree of differential equation. Techniques for solving first and second order linear and non-linear equations. Solutions of systems of first order linear equations. Finite linear differenceequations. Application to geometry and physics.

COS 201: **Computer Programming I** (3 Units, C: LH-30; PH-45)

Learning Outcomes

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

i. identify different programming paradigms and their approaches to programming;

ii. write programmes using basic data types and strings;

iii. design and implement programming problems using selection;

iv. design and implement programming problems using loops;

v. use and implement classes as data abstractions in an object-oriented approach;

vi. implement simple exception handling in programmes;

vii. develop programmes with input/output from text files; and

viii. design and implement programming problems involving arrays.

Course Contents

Introduction to computer programming. Functional programming; Declarative programming; Logic programming; Scripting languages. Introduction to object-orientation as a technique for modelling computation. Introduction of a typical object-oriented language, such as Java. Basicdata types, variables, expressions, assignment statements and operators. Basic object- oriented

concepts: abstraction; objects; classes; methods; parameter passing; encapsulation.Introduction to Strings and string processing; Simple I/O; control structures; Arrays; Simple recursive algorithms; inheritance; polymorphism.

Lab work: Programming assignments involving hands-on practice in the design and implementation of simple algorithms such as finding the average, standard deviation, searching and sorting. Practice in developing and tracing simple recursive algorithms. Developing programmes involving inheritance and polymorphism.

COS 202: **Computer Programming II** (3 Units, C: LH-30; PH-45)

Learning Outcomes

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

i. develop solutions for a range of problems using object-oriented programming;

ii. use modules/packages/namespaces for programme organisation;

iii. use API in writing applications;

iv. apply divide and conquer strategy to searching and sorting problems using iterative and/or recursive solutions;

v. explain the concept of exceptions in programming and how to handle exceptions in programmes;

vi. write simple multithreaded applications; and

vii. design and implement simple GUI applications.

Course Contents

This course is a continuation of CSC201. Review and coverage of advanced object-oriented programming - polymorphism, abstract classes and interfaces. Class hierarchies and programme organisation using packages/namespaces. Use of API – use of iterators/enumerators, List, Stack, Queue from API; Searching; sorting; Recursive algorithms;Event-driven programming: event-handling methods; event propagation; exception handling. Applications in Graphical User Interface (GUI) programming.

Lab work: Programming assignments leading to extensive practice in problem-solving and programme development with emphasis on object-orientation. Solving basic problems using static and dynamic data structures. Solving various searching and sorting algorithms using iterative and recursive approaches. GUI programming.

CSC 203: **Discrete Structures** (2 Units, C: LH-30)

Learning Outcomes

At the end of this course, the students will be able to:

i. convert logical statements from informal language to propositional and predicate logic expressions;

ii. describe the strengths and limitations of propositional and predicate logic;

iii. outline the basic structure of each proof technique (direct proof, proof by contradiction,

and induction) described in this unit;

iv. apply each of the proof techniques (direct proof, proof by contradiction, and induction) correctly in the construction of a sound argument;

v. apply the pigeonhole principle in the context of a formal proof;

vi. compute permutations and combinations of a set, and interpret the meaning in the contextof the particular application;

vii. map real-world applications to appropriate counting formalisms, such as determining the number of ways to arrange people around a table, subject to constraints on the seating arrangement, or the number of ways to determine certain hands in cards (e.g., a full house); and

viii. solve a variety of basic recurrence relations.

Course Contents

Propositional Logic. Predicate Logic. Sets. Functions. Sequences and Summation. Proof Techniques. Mathematical induction. Inclusion-exclusion and Pigeonhole principles. Permutations and Combinations (with and without repetitions). The Binomial Theorem. Discrete Probability. Recurrence Relations.

IFT 211: **Digital Logic Design** (2 Units,C: LH-15; PH-45)

Learning Outcomes

At the end of this course, students will be able to:

i. explain why everything is data, including instructions, in computers;

ii. describe how negative integers, fixed-length numbers, and non-numeric data are represented;

iii. convert numerical data from one format to another;

iv. describe computations as a system characterised by a known set of configurations with transitions from one unique configuration (state) to another (state);

v. describe the distinction between systems whose output is only a function of their input (combinational) and those with memory/history (sequential);

vi. describe a computer as a state machine that interprets machine instructions;

vii. articulate that there are many equivalent representations of computer functionality, including logical expressions and gates, and be able to use mathematical expressions to describe the functions of simple combinational and sequential circuits; and

viii. design the basic building blocks of a computer: arithmetic-logic unit (gate-level), registers (gate-level), central processing unit (register transfer-level), and memory (register transfer-level).

Course Contents

Introduction to information representation and number systems. Boolean algebra and switching theory. Manipulation and minimization of completely and incompletely specified Boolean

functions. Physical properties of gates: fan-in, fan-out, propagation delay, timing diagrams and tri-state drivers. Combinational circuits design using multiplexers, decoders, comparators and adders. Sequential circuit analysis and design, basic flip-flops, clocking andtiming diagrams. Registers, counters, RAMs, ROMs, PLAs, PLDs, and FPGAs.

Lab Work: Simple combinational gates (AND, OR, NOT, NAND, NOR); Combinational circuits design using multiplexers, decoders, comparators and adders. Sequential circuit analysis and design using basic flip-flops (S-R, J-K, D, T flip-flops); Demonstration of registers, counters, RAMs, ROMs, PLAs, PLDs, and FPGAs.

IFT 212: **Computer Architecture and Organization** (2 Units, C: LH-15; PH-45)

Learning Outcomes

At the end of this course, students will be able to:

i. explain the organization of the classical von Neumann machine and its major functional units;

ii. construct simple assembly language programme segments;

iii. describe how fundamental high-level programming constructs are implemented at the machine-language level;

iv. discuss the concept of control points and the generation of control signals using hardwired or micro programmed implementations;

v. describe how the use of memory hierarchy (cache, virtual memory) is used to reduce the effective memory latency; and

vi. explain the concept of interrupts and describe how they are used to implement I/O control and data transfers.

Course Contents

Principles of computer hardware and instruction set architecture. Internal CPU organization and implementation. Instruction format and types, memory, and I/O instructions. Dataflow, arithmetic, and flow control instructions, addressing modes, stack operations, and interrupts.Data path and control unit design. RTL, microprogramming and hardwired control. The practice of assembly language programming. Memory hierarchy. Cache memory, Virtual memory. Cache performance. Compiler support for cache performance. I/O organizations.

Lab work: Practical demonstration of the architecture of a typical computer. Illustration of different types of instructions and how they are executed. Simple Assembly Language programming. Demonstration of interrupts. Programming assignments to practice MS-DOS batch programming, Assembly Process, Debugging, Procedures, Keyboard input, VideoOutput, File and Disk I/O, and Data Structure. Demonstration of Reduced Instruction Set Computers. Illustration of parallel architectures and interconnection networks.

SEN 201: **Introduction to Software Engineering** (2 units, C: LH-30)

Learning Outcomes

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

i. describe the concept of the software life cycle;

ii. explain the phases of requirements analysis, design, development, testing andmaintenance in a typical software life cycle;

iii. differentiate amongst the various software development models;

iv. utilise UML for object-oriented analysis and design;

v. describe different design architectures;

vi. explain the various tasks involved in software project management; and

vii. describe the basic legal issues related to Software Engineering.

Course Contents

Software Engineering concepts and principles. Design, development and testing of software systems. Software processes: software lifecycle and process models. Process assessment models. Software process metrics. Life cycle of software system. Software requirements andspecifications. Software design. Software architecture. Software metrics. Software quality andtesting. Software architecture. Software validation. Software evolution: software maintenance; characteristics of maintainable software; re-engineering; legacy systems; software reuse. Software Engineering and its place as a computing discipline. Software projectmanagement: team management; project scheduling; software measurement and estimation techniques; risk analysis; software quality assurance; software configuration management. Software Engineering and law.

FUTM-CPT 211: **Back-End Web Development** (2 Units; C; LH-15, PH-45)

Learning Outcomes

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

i. implement an interactive web application using HTML, CSS, JavaScript , Bootstrap. and a wide range of web toolings.

ii. apply predefined CSS frameworks (Bootstrap 5) to styling website.

iii. write JavaScript programs for creating interactive websites.

iv. list and describe at least 5 tools needed for developing websites.

v. list and use at least 5 JavaScript Web API like Fetch API, Media API, Geolocation API etc.

Course Contents

Review of HTML and CSS. Styling website using predefined CSS frameworks such as Bootstrap 5. Creating dynamic website using JavaScript: Variables. Expression. Control structures. String methods. Arrays. Functions. Events and event handling. Client side form validation with JavaScript. Sending form data. POST and GET methods. JavaScript object and class. Working with JSON data. Introduction to Web API. API for manipulating document. Fetching data from server using Fetch API and XMLHttp Request. Drawing graphics. Video and audio API. Client side storage API. Asynchronous JavaScript: synchronous and asynchronous programming. Using promises. Promise-based APIs. Using Web Workers. Introduction to client-side JavaScript

frameworks. React. Ember. Vue. Svelte. Angular. Overview of client-side web development tools. Safety net tools. Transformation tools and post-deployment tools. Cross-browser testing.

Practical: Intensive practical sessions building dynamic and interactive client side web application with JavaScript. Develop graphic and animation with JavaScript, Use JavaScript for form validation etc. Create sites that with CSS Bootstrap framework.

FUTM-CPT 212: **Inferential Statistics for Computer Science** (2 Units; C; LH-15, PH-45)

Learning Outcome

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

i. make samples from a given population
ii. deduce conclusion on a given population from its sample
iii. differentiate between descriptive and inferential statistics
iv. list four probability distributions
v. apply different statistical tests as appropriate
vi. use SPSS in handling different data analysis.

Course Content

Population and Sample. Inferential Statistics. Probability. basic concepts. unconditional and conditional probabilities. Probability Distributions. normal distribution. chi square distribution. binomial distribution. and Poisson distribution. Confidence Interval. Hypothesis Testing. p-value. z-test. t-test. Chi-square test. ANOVA test. Use of SPSS to inferential statistics.

FUTM-CPT 221: **Numerical Computation for Computer Science** (2 Units; C; LH-15, P-45)

Learning Outcomes

On completion of the course, students should be able to:

i. state two methods for solving Systems of Linear Equations
ii. explain at least one Matrix factorization techniques
iii. compare Single Nonlinear Equation and Solving a System of Nonlinear Equations
iv. differentiate between the Runge-Kutta, explicit Runge-Kutta, and implicit Runge-Kutta methods
v. describe one MATLAB's interpolation tools
vi. state one method of data interpolation in python
vii. compare Trapezoid method and Simpson's method
viii. implement the numerical methods in MATLAB and Python

Course Contents

Solving Systems of Linear Equations: Direct method for solving linear systems. Testing the existence of the solution. Matrix factorization techniques. Iterative method and Ill-Conditioning and Regularization Techniques in Solutions of Linear Systems. Solving a System of Nonlinear

Equations- Solving a Single Nonlinear Equation and Solving a System of Nonlinear Equations. Interpolation and Solutions of Differential Equations. Lagrange interpolation. Newton's interpolation. MATLAB's interpolation tools, and data interpolation in python. Numerical Differentiation - Approximating Derivatives with Finite Differences. Numerical Integration: Trapezoid method. Simpson's method, Newton-Cotes Methods, and The Gauss Integration Method. Solving Systems of Nonlinear Ordinary Differential Equations- Runge-Kutta Methods. Explicit Runge-Kutta Methods. Implicit Runge-Kutta Methods, MATLAB ODE Solvers, and Python Solvers for IVPs. Nonstandard Finite Difference Methods for Solving ODEs- Deficiencies with Standard Finite Difference Schemes, Construction Rules of Nonstandard Finite Difference Schemes, Exact Finite Difference Schemes, and Other Nonstandard Finite Difference Schemes. Solving Optimization Problems- Linear and Quadratic Programming. Solving Optimization Problems- Nonlinear Programming. Solving Optimal Control Problems- The First-Order Optimality Conditions and Existence of Optimal Control. Necessary Conditions of the Discretized System. Numerical Solution of Optimal Control, and Solving Optimal Control Problems Using Indirect Methods.

FUTM-CPT 222: **Server-Side Web Development** (3 Units; C; LH-30, PH-45)

Learning Outcomes

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

i. Differentiate between front-end and Server-side web development.

ii. List at least 4 server-side scripting languages.

iii. Develop server-side computer programs using a modern server-side programming language like PHP and MySQL

iv. Develop server-side computer programs using PHP that access and manipulate a server-side relational database management system like MySQL.

v. Develop server-side computer programs that interact with external web services.

vi. Itemize at least 5 website security threats.

vii. State three tools and techniques for implementing web security

Course Content:

Review of HTML and CSS and JavaScript. Fundamental concepts of server-side programming. Overview of server-side scripting languages. PHP, Python, Ruby, C#, and JavaScript (NodeJS), ASP.net, JSP, Servlets. Basic PHP grammar and syntax. Variables,expression,controlmstructures. Arrays and string. PHP objects. Communication between server-side programs and client-side programs. Session management. Overview of SQL. Database programming with PHP and MySQL. SQL Create, Retrieve, Update, Delete statements. SQL querying. Accessing external web services. Developing a complete web application. Web applications security. Preventing SQL injection attacks. cross-site scripting (XSS), Cross-Site Request Forgery (CSRF). File inclusion. Directory traversal.

Practical will involve installation of webserver such as Apache webserver, MySQL database server to process server side scripts and store the database respectively. A full-fledged web application should be developed illustrating all the key principles taught in the course content.

FUTM-CPT 223: **System Concept and Design** (3 Units; C; L-30, P-45)

Learning Outcomes

On completion of the course, students should be able to:

i. state at least three fact gathering techniques
ii. describe four fundamental phases of the SDLC
iii. state three classes of system development methodologies
iv. describe two typical roles of system analyst
v. differentiate between use case diagrams. activity diagrams
vi. compare the merits of state chat diagram and sequence diagram
vii. state one way of managing project

Course Contents

Introduction to system concept. System Development Life Cycle (SDLC), Analysis - Fact gathering Techniques. Data flow diagrams. Process description. Data modelling. System Design - Structure Charts, form designs, security automated tools for design. The four fundamental phases of the SDLC and the steps involved. System Development Methodologies. The three different classes of system development methodologies. Typical System Analyst Roles and skills. Business Analyst, Systems Analyst, Change Management Analyst and Project Manager. The Unified Modelling Language (UML) -Models and use case diagrams. Activity Diagrams and State Chat Diagrams. Sequence and Collaboration Diagram, Class Diagrams. Component Diagrams. Customers, organization types, project management, teams and team dynamics. Computer Assisted Software Engineering (CASE) tools. Documentation. Importance of CASE tools.

Lab Work: Analysis and design assignments leading to extensive practice in the use of UML and CASE tools.

300 Level

GST 312: **Peace and Conflict Resolution** (2 Units, C: LH-30)

Learning Outcomes

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

i. analyse the concepts of peace, conflict and security;
ii. list major forms, types and root causes of conflict and violence;
iii. differentiate between conflict and terrorism;
iv. enumerate security and peacebuilding strategies; and
v. describe roles of international organisations, media and traditional institutions in peacebuilding.

Course Contents

Concepts of Peace, Conflict and Security in a multi-ethnic nation. Types and Theories of Conflicts: Ethnic, Religious, Economic, Geopolitical Conflicts; Structural Conflict Theory, RealistTheory of Conflict, Frustration-Aggression Conflict Theory. Root causes of Conflict and Violence in Africa: Indigene and Settlers Phenomenon; Boundaries/border disputes; Political disputes; Ethnic disputes and rivalries; Economic Inequalities; Social disputes; Nationalist Movements and Agitations; Selected Conflict Case Studies – Tiv-Junkun; Zango Kartaf, Chieftaincy and Land disputes, etc. Peace Building, Management of Conflicts and Security: Peace & Human Development. Approaches to Peace & Conflict Management (Religious, Government, Community Leaders, etc.). Elements of Peace Studies and Conflict Resolution: Conflict dynamics assessment Scales: Constructive & Destructive. Justice and Legal framework: Concepts of Social Justice; The Nigeria Legal System. Insurgency and Terrorism. Peace Mediation and Peace Keeping. Peace & Security Council (International, National and Local levels) Agents of Conflict resolution – Conventions, Treaties Community Policing: Evolution and Imperatives. Alternative Dispute Resolution, ADR. Dialogue b). Arbitration, c). Negotiation d). Collaboration, etc. Roles of International Organisations in Conflict Resolution.(a). The United Nations, UN and its Conflict Resolution Organs. (b). The African Union & PeaceSecurity Council (c). ECOWAS in Peace Keeping. Media and Traditional Institutions in Peace. Building. Managing Post-Conflict Situations/Crisis: Refugees. Internally Displaced Persons,IDPs. The role of NGOs in Post-Conflict Situations/Crisis.

ENT 312: **Venture Creation** (2 Units, C: LH-15; PH-45)

Learning Outcomes

At the end of this course, students, through case study and practical approaches, should be able to:

i. describe the key steps in venture creation;
ii. spot opportunities in problems and in high potential sectors regardless of geographical location;
iii. state how original products, ideas, and concepts are developed;
iv. develop business concept for further incubation or pitching for funding;
v. identify key sources of entrepreneurial finance;
vi. implement the requirements for establishing and managing micro and small enterprises;
vii. conduct entrepreneurial marketing and e-commerce;
viii. apply a wide variety of emerging technological solutions to entrepreneurship; and
ix. appreciate why ventures fail due to lack of planning and poor implementation.

Course Contents

Opportunity Identification (Sources of business opportunities in Nigeria, Environmental scanning, Demand and supply gap/unmet needs/market gaps/market research, Unutilised resources, Social and climate conditions, and technology adoption gap). New business development (business planning, market research). Entrepreneurial finance (venture capital, equity finance, microfinance,

personal savings, small business investment organisations, and business plan competition). Entrepreneurial marketing and e-commerce (Principles of marketing, customer acquisition & retention, B2B, C2C and B2C models of e-commerce, first mover advantage, e-commerce business models and successful e-commerce companies,). Small business management/family business: Leadership & Management, basic bookkeeping,nature of family business and family business growth model. Negotiation and business communication (Strategy and tactics of negotiation/bargaining, traditional and modern business communication methods). Opportunity discovery demonstrations (business idea generation presentations, business idea contest, brainstorming sessions, idea pitching). Technological solutions (the concept of market/customer solution, customer solution, and emerging technologies, business applications of new technologies- Artificial Intelligence (AI),Virtual/Mixed Reality (VR), Internet of Things (IoT), Blockchain, Cloud Computing, renewableenergy, etc. digital business and e-commerce strategies).

CSC 301: **Data Structures** (3 Units, C: LH-30; PH-45)

Learning Outcomes

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

i. discuss the appropriate use of built-in data structures;

ii. apply object-oriented concepts (inheritance, polymorphism, design patterns, etc.) in software design;

iii. implement various data structures and their algorithms, and apply them in implementing simple applications;

iv. choose the appropriate data structure for modelling a given problem;

v. analyse simple algorithms and determine their efficiency using big-O notation; and

vi. apply the knowledge of data structures to other application domains like data compression and memory management.

Course Contents

Primitive types, Arrays, Records Strings and String processing. Data representation in memory, Stack and Heap allocation, Queues, Trees. Implementation strategies for stack, queues, trees.Run time storage management; Pointers and References, linked structures.

Lab work: Writing C^{+}/C^{++} functions to perform practical exercises and implement using the algorithms on arrays, records, string processing, queues, trees, pointers and linked structures.

CSC 308: **Operating System** (3 Units, C: LH-30; PH-45)

Learning Outcomes

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

i. recognise operating system types and structures;

ii. describe OS support for processes and threads;

iii. recognise CPU scheduling, synchronisation, and deadlock;

iv. resolve OS issues related to synchronisation and failure for distributed systems;

v. explain OS support for virtual memory, disk scheduling, I/O, and file systems;

vi. identify security and protection issues in computer systems; and

vii. use C and Unix commands, examine behaviour and performance of Linux, and develop various system programmes under Linux to make use of OS concepts related to process synchronisation, shared memory, mailboxes, file systems, etc.

Course Contents

Fundamentals of operating systems design and implementation. History and evolution of operating systems. Types of operating systems. Operating system structures. Process management: processes, threads, CPU scheduling, process synchronisation. Memory management and virtual memory. File systems; I/O systems; Security and protection; Distributed systems; Case studies.

Lab work: Practical hands-on engagement to facilitate understanding of the material taughtin the course. All the process, memory, file and directory management issues will be demonstrated under the LINUX operating system. Also UNIX commands will be briefly discussed. Alternatively, hands-on exposure may be through the use of operating systems developed for teaching, like TempOS, Nachos, Xinu or MiniOS. Another possibility is through programming exercises that implement and simulate algorithms taught. Simulation of CPU scheduling algorithms, producer-consumer problem, memory allocation algorithms, file organisation techniques, deadlock algorithms and disk scheduling algorithms.

CSC 309: **Artificial Intelligence** (2 Units, C: LH-15; PH-45)

Learning Outcomes

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

i. explain AI fundamentals, concepts, goals, types, techniques, branches, applications, AI technology and tools;

ii. discuss intelligent agents, their performance, examples, faculties, environment and architectures, and determine the characteristics of a given problem that an intelligent system must solve;

iii. describe the Turing test and the “Chinese Room” thought experiment, and differentiate between the concepts of optimal reasoning/behaviour and human-like reasoning/behaviour;

iv. describe the role of heuristics and the trade-offs among completeness, optimality, time complexity, and space complexity;

v. analyse the types of search and their applications in AI and describe the problem of combinatorial explosion of search space and its consequences;

vi. demonstrate knowledge representation, semantic network and frames along with their applicable uses;

vii. practice Natural Language Processing, translate a natural language (e.g., English) sentence

into a predicate logic statement, convert a logic statement into clause form, apply resolution to a set of logic statements to answer a query; and

viii. analyse programming languages for AI and expert systems technology, and employ application domains of AI.

Course Contents

Overview of Artificial Intelligence. History of AI. Goals of AI. AI Technique. Types of AI. Branches and applications of AI. Advantages and Disadvantages. Introduction to Intelligent Agents. Agent Performance, Examples of Agents, Agent Faculties, Rationality, Agent Environment. Agent Architectures. Search. General Classes of AI Search Algorithm Problems. Problem Solving by Search. Types of AI Search Techniques and Strategies. Introduction to thetypes of problems and techniques in AI. Problem-Solving methods. Major structures used in AI programmes. Knowledge Representation. KR and Reasoning Challenges. KR Languages. Knowledge representation techniques such as predicate logic, non-monotonic logic, and probabilistic reasoning. Semantic Network - types of relationships, semantic network inheritance, types and components. Introduction to Frames. Natural Language Processing (NLP). Introduction to natural language understanding and various syntactic and semantic structures. Introduction to Expert Systems - characteristics, components, types, requirements, technology, development. Programming Languages for AI. Introduction to computer image recognition.

Lab work: Group practical in (i) Turing test practical - Students can act out their own versionof the Turing test (ii) Facial recognition practical to aid in teaching students how machine learning works with students simulating a facial recognition algorithm. Practical applications of NLP in groups – (i) Question Answering focuses on building systems that automatically answer the questions asked by humans in a natural language (ii) Spam detection applicationfor detecting unwanted e-mails getting to a user's inbox (iii) Sentiment analysis/opinion miningshould be used on the web to analyse the attitude, behaviour, and emotional state of the sender, implemented through a combination of NLP and statistics (iv) Practical exercise of machine translation used to translate text or speech from one natural language to another natural language such as the Google Translator (v) Developing a model to provide word processor software for the spelling correction (vi) Developing a model for speech recognition for converting spoken words into text (vii) Implementing a Chatbot to provide the staff/student's chat services. OR

Group Practical exercise on agents and its environment using simulation of a colony of ants foraging for food; model simulating a message between agents; model simulating the flocking behaviour of birds; model to apply standard search algorithm to the classic search problem of missionaries and cannibals, and how to use communicating agents for searching networks.

Some computer AI animation exercises for any branch of AI. Practical exercise on simple robots coupling and programming. Group project of building a lawn robot for trimming grasses, or any simple design and implementation of robotics.

CSC 322: **Computer Science Innovation and New Technologies** (2 Units, C: LH-30)

Learning Outcomes

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

i. explain business models;

ii. identify some entrepreneurial opportunities available in IT;

iii. describe business plan and business startup process;

iv. explain business feasibility and strategy;

v. explain marketing strategies; and

vi. discuss business ethics and legal issues.

Course Contents

Fundamental concepts of innovation and business ideas in general. Product development. Business leadership. Digital marketing. Entrepreneurial opportunities in IT. Legal issues and Business ethics. New venture creation process. Business feasibility planning. Market research. Business strategy. Business models and Business plans. Technical presentations. Report on asuccessful entrepreneurial outfit.

CYB 201: **Introduction to Cybersecurity and Strategy** (2 Units, C: LH-30)

Learning Outcomes

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

i. explain cybersecurity concepts, its methods, elements, and terminologies of cyber security

ii. cyber, security, threat, attack, defence, and operations;

iii. describe common cyber-attacks and threats, cybersecurity issues, challenges and proffered solutions, and build an enhanced view of main actors of cyberspace and cyber operations;

iv. apply the techniques for identifying, detecting, and defending against cybersecurity threats, attacks and protecting information assets;

v. explain the impact of cybersecurity on civil and military institutions, privacy, business and government applications;

vi. identify the methods and motives of cybersecurity incident perpetrators, and the countermeasures employed by organisations and agencies to prevent and detect those incidences and software application vulnerabilities; and

vii. state the ethical obligations of security professionals, evaluate cybersecurity and national security strategies to the typologies of cyber-attacks that require policy tools and domestic response, and define the cybersecurity requirements and strategies evolving in the face of big risk.

Course Contents

Basic concepts: cyber, security, confidentiality, integrity, availability, authentication, access control, non-repudiation and fault-tolerant methodologies for implementing security. Security policies, best current practices, testing security, and incident response, Risk management, disaster recovery and access control. Basic cryptography and software application vulnerabilities. Evolution of cyber-attacks. Operating system protection mechanisms, intrusiondetection systems, basic formal models of security, cryptography, steganography, network and distributed system security, denial of service (and other) attack strategies, worms, viruses, transfer of funds/value across networks, electronic voting, secure applications. Cybersecurity policy and guidelines.

Government regulation of information technology. Main actors of cyberspace and cyber operations. Impact of cybersecurity on civil and military institutions, privacy, business and government applications; examination of the dimensions ofnetworks, protocols, operating systems, and associated applications. Methods and motives ofcybersecurity incident perpetrators, and the countermeasures employed by organisations andagencies to prevent and detect those incidences. Ethical obligations of security professionals.Trends and development in cybersecurity. Software application vulnerabilities. Evolution of cybersecurity and national security strategies, requirements to the typologies of cyber-attacksthat require policy tools and domestic response. Cybersecurity strategies evolving in the faceof big risk. Role of standards and frameworks.

DTS 304: **Data Management I** (3 Units, C: LH-30; PH-45)

Learning Outcomes

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

i. describe the components of a database system and give examples of their use;

ii. describe the differences between relational and semi-structured data models;

iii. explain and demonstrate the concepts of entity integrity constraint and referential integrity constraint;

iv. apply queries, query optimisations and functional dependencies in relational databases;

v. describe properties of normal forms and explain the impact of normalisation on the efficiency of database operations;

vi. describe database security and integrity issues and their importance in database design; and

vii. explain the concepts of concurrency control and recovery mechanisms in databases.

Course Contents

Information Management Concepts. Information storage & retrieval. Information management applications. Information capture and representation. Analysis and indexing - search, retrieval, information privacy. Integrity and security. Scalability, Efficiency and Effectiveness. Introduction to database systems. Components of database systems. DBMS functions. Database architecture and data independence. Database query language. Conceptual models. Relational data models. Semi-structured data models. Relational theory and languages. Database Design. Database security and integrity. Introduction to query processing and optimisation. Introduction to concurrency and recovery.

Lab work: Practical exercise on information representation, capture, storage and retrieval. Learn how to analyse data and index for easy searching and indexing. Practical on creating database files and models. How to create and use various database designs. How to query the created database. Methods of concurrency and recovery in database. Learn how to securethe database.

ICT 305: **Data Communication Systems and Network** (3 Units C: LH-30; PH-45)

Learning Outcomes

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

i. explain data transmission over layered networks;

ii. list and explain common internet technologies and protocols; and

iii. explain network operating system.

Course Contents

Types and sources of data. Simple communications network. Transmission definitions, one way transmission, half duplex transmission, transmission codes, transmission modes, parallel transmission, serial transmission, bit synchronisation, character synchronisation, synchronous transmission, asynchronous transmission, efficiency of transmission. Introduction to network protocol. Seven Layer ISO-OSI standard protocols and network architecture. Transport protocols, session services protocols, and other protocols. Institute of Electrical and ElectronicsEngineering 802 standards. Error control and Data Compression: Forward Error Control; error detection methods; parity checking; linear block codes, cyclic redundancy checking; feedbackerror control, data compression, Huffman coding and dynamic Huffman coding. Local Area Networks: medium access control techniques – Ethernet, token bus and token ring; fibre distributed data interface, metropolitan area network. Peer-to-peer, Client Server. Client- Server Requirements: GUI design standards, interface independence, platform independence,transaction processing, connectivity, reliability, backup, and recovery mechanisms. Features and benefits of major recovery mechanisms. Network OS: (e.g., Novell NetWare, UNIX/LINUX, OS/2 & Windows NT). INTERNET: Definition, architecture, services, internet addressing. Internet protocol, IPv4, IPv6.

Lab Work: Demonstration of simple communications networks. Illustration of applications atthe various levels of the OSI model. Demonstration of different types of Local Area Networks(LANs). Illustration of Metropolitan Area Networks. Illustration of Error Detection and Error Correction techniques. Demonstration of Network Operating Systems.

FUTM-CPT 311: **Programming Language Translation and Compiler Design** (3 units; C; LH-45, PH-0)

Learning Outcomes

On completion of the course, students should be able to:

i. state at least two compilers

ii. describe two interpreters

iii. state two grammars and languages

iv. describe one syntax semantics

v. differentiate between assemblers and interpreters

vi. compare lexical and expression analysis

Course Content

Introduction to programming language translation and compiler design. Compilers, assemblers and interpreters. Structure and functional aspects of a typical compiler. Syntax semantics and functional relationship between lexical analysis, expression, analysis and code generation.

Internal form of course programme. Error detection and recovery. The parsing problem and the scanner. Grammars and languages. Recognizers, Top-down and bottom-up language. Run-time storage Organization. The use of display in run-time storage Organization. The use of display in run time storage. Allocation LR grammars and analyzers. Construction of LR table. Organization of symbol tablets. Allocation of storage to run-time variables. Code generation and Optimization/Translator with systems.

FUTM-CPT 312: **Object-Oriented Analysis, Design and Implementation** (2 Units; C; LH-15, PH-45)

Learning Outcomes

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

i. describe the 4 fundamental concepts of object orientation.

ii. state at least three main features of the software development process in an object-oriented framework.

iii. describe one Visual Object Oriented Modelling language, specifically UML (Unified Modeling Language) for designing software.

iv. read, verify, and validate a given specification presented in UML

v. transform a given a system requirements description to produce a specification using UML.

vi. implement software from OOD represented with UML.

vii. differentiate between use case, class diagram, sequence diagram and other UML diagrams.

Course Content

Object Oriented Fundamentals: Definition of Object Oriented Analysis and Design. Defining Models. Requirement Process. Object Oriented Development Cycle. Overview of the Unified Modeling Language. UML Fundamentals and Notations. Object Oriented Analysis: Building Conceptual Model. Adding Associations and Attributes. Representation of System Behavior. Object Oriented Design: Analysis to Design. Describing and Elaborating Use Cases. Collaboration Diagram. Objects and Patterns. Determining Visibility. Class Diagram. Implementation: Programming and Development Process. Mapping Design to Code. Creating Class Definitions from Design Class Diagrams. Creating Methods from Collaboration Diagram. Updating Class Definitions. Classes in Code. Exception and Error Handling.

Practical:

Laboratory Exercise will include handling a object oriented design and modelling activity in a ACSE Environment. UML pattern design and modelling will be taken up with the help of UML Software and implement the design using object oriented language like Java or C++.

FUTM-CPT 321: **Human Computer Interaction** (3 Units; C; LH-30, PH-45)

Learning Outcomes

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

i. discuss the foundations and concept of human-computer interface;

ii. explain the principles of human-computer interface;

iii. explain the design and development of the human-computer interface; and

iv. explain the importance of user feedback.

v. know how to carry out evaluation of different types of software

Course Contents

Foundations of HCI. Concept underlying the design of HCI. Principles of GUI. GUI toolkits. System design methods. User conceptual models and interface metaphors. Human cognitive and physical ergonomics. Human-centred software evaluation and development. GUI design and programming.

FUTM-CPT 322: **Theory of Computing** (3 Units; C: L-30, PH-45)

Learning Outcomes

On completion of the course, students should be able to:

i. state at least two types of automata

ii. describe three applications of automaton to fields of computing.

iii. state three kinds of automata recognizable languages

iv. describe two concepts of Regular Expressions

v. differentiate between deterministic and non-deterministic automata

vi. compare Time, Space and Probability complexity

Course Content

Introduction to theory of computing. Finite automata, Regular languages and Regular expressions. Deterministic finite automata and nondeterministic finite automata. Regular expressions and FA-recognisable languages. Non-regular languages. Algorithms that answer questions about FAs and regular expressions. Computability theory. Turin machines, and nondeterministic Turing machines. Undecidability, PCP, counter and stack machines. Reducibility and recursion theorem. Complexity Theory. Time complexity, nondeterministic time complexity, P and NP. NP-completeness III. Poly-time reductions and Cook-Levin theorem. NP-completeness II and space complexity III. Probabilistic complexity. Probabilistic complexity. Interactive proofs.

FUTM-CPT 323: **Mobile Application Development for the Android Platform** (3 Units; C; LH-30, PH-45)

Learning Outcomes

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

i. list and use the four major components of Android Application for creating apps.

ii. use Android Studio to build Android apps.

iii. create and design user interfaces for Android apps using Views, Layouts, and other UI components.

iv. store and retrieve data in Android apps, including local storage SharedPreferences and

SQLite databases.

v. list the steps to follow to publish apps on the Google Play Store.

Course Content

Introduction to Android development. Overview of the Android operating system. .The Android SDK. and the Android development environment. Programming fundamentals. Overview of programming concepts and syntax. such as variables. functions. and data structures. with a focus on Java and Kotlin.. Android app architecture. Introduction to the components and architecture involved in building an Android app. including activities. services. broadcast receivers. and content providers. User interface design. Overview of user interface (UI) design for Android apps including layout managers. UI widgets. and resources. Use of Jetpack compose for UI design. Android jetpack libraries. Data storage. Overview of data storage options for Android apps. including SQLite databases. shared preferences. and files. Networking. Overview of networking concepts and implementation in Android. including web services and APIs Advanced topics in Android development. Animation .Multimedia. Sensors. Maps and location services. Testing and deployment. Overview of testing and deployment of Android apps Use of tools and services for testing. debugging. and publishing Android apps on the Google Play Store.

Practical: Demonstration of a Simple Mobile Application. Design and Development of interactive mobile applications. Demonstration of multiplatform mobile application development. Development of Android applications including UI design and data storage design. Demonstration of advanced mobile application design. Illustration of metrics for measuring the performance of mobile applications.

400 Level

COS 409: **Research Methodology and Technical Report Writing** (3 Units, C: LH-45)

Learning Outcomes

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

i. distinguish qualitative and quantitative research methodologies and their applications;

ii. identify and define a research problem in a given area;

iii. identify different methods of data collection and select the methods appropriate to a given situation;

iv. design and conduct simple research including analysis and interpretation of research results;

v. document research problem, methodology all the way to research report writing;

vi. defend the written research report; and

vii. familiarise themselves with ethical issues in the conduct of research.

Course Contents

Foundations of Research. Types of Research. Research Approaches. Significance of Research. Research Methods versus Methodology. Research Process. Criteria and Strategy for Good Research. Problems Encountered by Researchers in Nigeria. Principles of Scientific Research.

Scientific investigation. Problem formulation. Definition and technique of the ResearchProblem. Selection of Appropriate Method for Data Collection- Primary Data and Secondary Data. Guidelines for Constructing Questionnaire/Schedule. Guidelines for Successful Interviewing. Difference between Survey and Experiment. Eloping Research Proposal and Research Plan. Formulation of working hypothesis and Testing. Literature review. Procedure for reviewing related relevant studies and referencing cited works. Types of Reports. TechnicalReport Writing. Layout and mechanics of writing a Research Report. Standard Techniques forResearch Documentation. Sampling Design. Different Types of Sample Designs. Steps in Sampling Design. Criteria of Selecting a Sampling Procedure. Methods of analysis. Processing and Analysis of Data Elements/Types of Analysis. Interpretation and Presentation of results. How to prepare References and Bibliography.

CSC 401: **Algorithms and Complexity Analysis** (2 Units, C: LH-30)

Learning Outcomes

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

i. explain the use of big-O, omega, and theta notation to describe the amount of work doneby an algorithm,

ii. use big-O, omega, and theta notation to give asymptotic upper, lower, and tight boundson time and space complexity of algorithms,

iii. determine the time and space complexity of simple algorithms,

iv. deduce recurrence relations that describe the time complexity of recursively defined algorithms,

v. solve elementary recurrence relations,

vi. for each of the strategies (brute-force, greedy, divide-and-conquer, recursive backtracking, and dynamic programming), identify a practical example to which it would apply,

vii. use pattern matching to analyse substrings, and

viii. use numerical approximation to solve mathematical problems, such as finding the roots ofa polynomial.

Course Contents

Basic algorithmic analysis. Asymptotic analysis of Upper and average complexity bounds. Standard Complexity Classes. Time and space trade-offs in analysis recursive algorithms. Algorithmic Strategies. Fundamental computing algorithms. Numerical algorithms. Sequentialand Binary search algorithms. Sorting algorithms, Binary Search trees. Hash tables. Graphs and their representation.

CSC 402: **Ethics and Legal Issues in Computer Science** (2 Units, C: LH-30)

Learning Outcomes

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

i. state laws and regulations related to ethics;

ii. identify and explain relevant codes of ethics for computing practice;

iii. identify social and ethical issues in different areas of computing practice;

iv. review real-life ethical cases and be able to develop ethical resolutions and policies;

v. explain the consequences of ignoring and non-compliance with ethical provisions; and

vi. develop a sound methodology in resolving ethical conflicts and crisis.

Course Contents

Addresses social, ethical, legal and managerial issues in the application of Computer Scienceto the information technology industry. Through seminars and case studies, human issues confronting Computer Science graduates will be addressed. Topics include managerial and personal ethics, computer security, privacy, software reliability, personal responsibility for the quality of work, intellectual property, environment and health concerns, and fairness in the workplace.

INS 401: Project Management (2 Units, C: LH-30)

Learning Outcomes

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

i. describe project management planning;

ii. describe project scheduling;

iii. explain management of project resources;

iv. discuss project procurement, monitoring and execution; and

v. explain project communication and time management.

Course Contents

Introduction to Project Management. The Project Management Lifecycle: Project managementand systems development or acquisition. The project management context. Technology and techniques to support the project management lifecycle, and Project management processes.Managing Project Teams: Project team planning, motivating team members, Leadership, power and conflict in project teams, and managing global project teams. Managing project communication and enhancing team communication. Project Initiation and Planning. Managing Project Scope: Project initiation, how organisations choose projects, Activities, and developing the project charter. Managing Project Scheduling: Common problems in project scheduling, and Techniques for project scheduling. Managing Project Resources: Types of resources (human, capital, time), and Techniques for managing resources. Project quality andtools to manage project quality. Managing project risk and tools for managing project risk. Managing Project Procurement: Alternatives to systems development, External acquisition, Outsourcing-domestic and offshore. Steps in the procurement process, and managing the procurement process. Project Execution, Control and Closure: Managing project execution, monitoring progress and managing change. Documentation and communication, and Commonproblems in project execution. Managing Project Control and Closure: Obtaining information,Cost control, Change control, administrative closure, Personnel closure, Contractual closure and Project auditing.

FUTM-CPT 411: **Introduction to Machine Learning for Data Mining** (3 Units; C; L-30, P-45)

Learning Outcomes

On completion of the course, students should be able to:

i. state at least two tasks of data mining
ii. describe two data pre-processing and preparation methods
iii. state three Neural Network algorithms
iv. describe two performance metrics use to evaluate models based on the domain of task
v. differentiate between classification and prediction,
vi. compare regression, anomalous detection and clustering on sampled data
vii. apply Classification and regression analysis with Artificial Neural Network

Course Contents

Overview of data mining. Data pre-processing descriptive data summarization. Data cleaning. Data integration and transformation. Data reduction. Data discretization and concept hierarchy generation. Data preparation. Overview. Cleaning the data. Removing variables. Data transformation. Segmentation. Table and graphs. Tables. Data tables. Contingency tables. Graphs. Frequency. Polygram and histograms. Scatter plots. Box plots. Multiple graphs. Prediction – classification. Regression 1. Building and applying a prediction model. Prediction – classification. Regression 2. Predicting using decision trees. Naive Bayes estimation and Bayesian networks. Posterior odds ratio. Balancing the data. Prediction. Classification. Regression 3. Naive Bayes classification. numeric predictors analysis using Naive Bayes. Bayesian belief networks. Cloth purchase example. using the Bayesian Network to find probabilities. Prediction. Classification. Regression 4. Genetic Algorithm Introduction. Basic framework of a ga. Simple example of a genetic algorithm. Cross over. Multipoint crossover. Uniform crossover. Analysis using genetic algorithm. Association rules and cluster analysis. Basic Concepts. Efficient and Scalable Frequent Item. Set Mining Methods. Mining. Various Kinds of Association Rules. Cluster Analysis/ Types of Data in Cluster Analysis. A Categorization of Major Clustering Methods. Different Clustering Methods. Classification and Prediction: Classification. Issues Regarding Classification and Prediction. Different classifications. Classification by decision tree induction. Bayesian classification. Rule based classification. Classification by back propagation. Prediction. Accuracy and error measures. Evaluating the accuracy of a classifier or predictor. Ensemble methods. Model selection. Various minings. Mining data streams. Mining time series data. Data graph mining.social network analysis. Multi.relational data mining. Multimedia mining and applications: multidimensional analysis and descriptive mining of complex data objects. Spatial data mining. Multimedia data mining. Text mining. Mining the www.applications and trends in data mining. Overview of neural networks: basic architecture of neural networks and neural computing.

FUTM-CPT 412: **Advanced Visual Programming with VB.net** (3 Units, C; LH-30, PH-45)

Learning Outcomes

At the end of this course Students should be able to:

i. design, create, build, and debug Visual Basic applications.

ii. explore Visual Basic's Integrated Development Environment (IDE).

iii. explain variables and data types used in program development.

iv. write and apply procedures, sub-procedures, and functions to create manageable code.

v. create one- and two-dimensional arrays for sorting, calculating, and displaying of data.

vi. write Visual Basic programs using object-oriented programming techniques including classes, objects, methods, instance variables, composition, and inheritance, and polymorphism.

Course contents

Introduction: Windows concepts. Objects and events. define design and development process. identify elements of ID. Introduce More Controls and Their Properties. Variables. Constants. and Calculations. Decisions and Conditions. Menus. Procedures and Functions. The .NET framework: Visual Basic 2010 Productivity features. Exceptions and Events: designing and consuming events; structured exception handling. Data manipulation with ADO.NET. Applications. Control Statements. Arrays. lasses. Objects. Methods and Instance Variables. . Use the Object Browser to navigate the .NET Framework Class Library. Declaring Methods with Parameters. Instance Variables and Properties. Value Types and Reference Shared/Class Methods vs. Instance Methods. Subroutine vs. Function Methods. Declaring and Using Methods. Passing Arguments: Pass-by-Value vs. Pass-by-Reference. Method Overloading. Optional parameters. Recursion. Scope. Constructors. Composition. Inheritance. Forms and Controls: Labels. Textboxes. Buttons. Checkboxes. Dialog Boxes. Combo Boxes. Radio Buttons.

CSC 499: **SIWES I & II** (6 Units, C: PH-270)

Learning Outcomes

At the end of this training, students should be able to:

i. explain how a typical computer firm/unit operates;

ii. describe the various assignments carried out and the skills acquired during the SIWES period; and

iii. submit a comprehensive report on the knowledge acquired and the experience gained during the exercise.

Course Contents

Students are attached to private and public organisations for a period of three months duringthe second-year session long break with a view to making them acquire practical experience and to the extent possible, develop skills in all areas of Computer Science. Students are supervised during the training period and shall be expected to keep records designed for thepurpose of monitoring their performance. They are also expected to submit a report on the experience gained and defend their reports.

FUTM-CPT 511: **Introduction to Cloud Computing** (3 Units; C; LH-30, PH-45)

Learning Outcomes

On completion of the course, students should be able to:

i. state at least three cloud computing models
ii. describe three cloud computing infrastructures
iii. state three security and privacy issues of cloud computing
iv. describe one cloud computing architecture
v. differentiate between private and public cloud
vi. compare the merits of cloud infrastructure

Course Contents

Cloud computing overview. Cloud computing deployment models. Security and privacy of cloud computing. Cloud computing planning. Cloud computing technologies. Cloud computing architecture. Cloud computing infrastructure. Public cloud model. Private cloud model. Hybrid cloud model. Community cloud model. Infrastructure-as-a-Service (IaaS). Platform-as-a-Service (PaaS). Software-as-a-Service (SaaS). Identity-as-a-Service. Network-as-a-Service (NaaS). Cloud management.

FUTM-CPT 512: **Operations Research** (3 Units; C; LH-45, PH-0)

Learning Outcome

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

i. define and know the meaning of Operations Research (OR)
ii. know how to use the tools of OR
iii. appreciate and apply OR in making decisions
iv. know what OR can be used for and it cannot be used for.
v. use OR in modelling systems
vi. solve OR problem using different methods
vii. know how business managers and OP specialist relate.

Course Content

Introduction to Operations Research; elementary concepts and objectives of operations research. applications of operations research in decision making. Linear Programming Problem; mathematical formulation of the linear programming problem and its graphical solution. simplex method. Transportation Problems; definition and mathematical formulation. initial basic feasible solution. optimal solution. Assignment Problem; introduction and mathematical formulation. solution of assignment problems. Inventory Control; introduction and general notations. economic lot size models with known demand. Replacement Theory; introduction and elementary concepts. replacement of items deteriorating with time. Sequencing Problem; introduction and general notations. solution of a sequencing problem. Queuing Theory; introduction and classification of queues. solution of queuing models. Project Planning and Network Analysis; introduction and basic definitions in network analysis. rules for drawing network analysis. critical path method (CPM). project evaluation and review technique (PERT).

FUTM-CPT 513: **Introduction to Applied Neural Network and Deep Learning** (3 Units; C; LH-15, PH-45)

Learning Outcomes

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

i. list at least 4 deep learning methods covered in the course, including the basic concepts, the key algorithms, and the commonly-used implementation of the methods.

ii. identify societal challenges that can potentially be tackled by deep learning methods, and determine which specific methods can be applied

iii. formulate a specific societal problem in a way that it is amenable to deep learning solutions and propose how to adjust and modify the deep learning techniques to fit the problem.

iv. implement basic convolution neural network.

v. implement basic recurrent neural network

vi. use existing deep learning libraries like tensorflow and keras to implement deep learning systems.

vii. state the principle of AI Ethics and responsible use of AI for social good.

Course Contents

Learning neural networks for Classification and Regression. Backpropagation Algorithm. Gradient Descent Optimization Techniques and Its variations. Deep Neural Network. Convolutional Neural Network. Convolution operation. Pooling layer. Max-pooling. Recurrent Neural Network.Basic RÑN. Long term short memory (LSTM). Gated recurrent network (GRU). Generative Model. Auto encoders. Generative Adversarial Network (GAN). Restricted Boltzmann Machine RBM Deep Reinforcement Learning. Application of Deep Learning to Social Good: Case studies of societal problems that DL methods can be applied to, Introduction to machine learning problem framing, sourcing data, AI ethics.

Practical: Python based deep learning libraries such Keras and Tensorflow lwill be used for practical implementation and testing of the various methods taught in the course. Student will be given term paper to identify and solve a societal problem using the methods studied in the course.

FUTM-CPT 514: **Introduction to Natural Language Processing** (3 Units; C; LH-15, PH-45)

Learning Outcomes

On completion of this course student should be able to:

i. list and use at least 5 techniques of NLP, including text processing, tokenization, part-of-speech tagging, named entity recognition, and more.

ii. develop hands-on experience with NLP tools and techniques using programming languages such as Python.

iii. analyze and process text data, including how to extract information from unstructured text and categorize text into predefined categories.

iv. list and develop solutions for NLP tasks such as sentiment analysis, text classification,

machine translation, and word embeddings.

v. develop critical thinking and problem-solving skills in NLP.

vi. execute further research and development in NLP and related fields by gaining a comprehensive understanding of NLP concepts and techniques.

vii. identify the impact and ethical considerations of NLP, including the potential benefits and risks associated with its use in various domains and industries.

Course Contents

Language Models. The bag-of-words model. N-gram word models. Other n-gram models. Smoothing n-gram models. Word representations. Part-of-speech (POS) tagging. Comparing language models. Grammar - The lexicon of E_0. Parsing .Dependency parsing. Learning a parser from examples. Augmented Grammar. Semantic interpretation. Learning semantic grammars. Complications of Real Natural Language .Natural Language Tasks. Deep Learning for Natural Language Processing. Word Embedding. Recurrent Neural Networks for NLP. Language models with recurrent neural networks. Classification with recurrent neural networks. LSTMs for NLP tasks. Sequence-to-Sequence Models. Attention. Decoding. The Transformer Architecture. Self-attention. From self-attention to transformer. Pretraining and Transfer Learning. Pretrained word embedding. Pretrained contextual representations. Masked language models.

Practical: Python based deep learning libraries such Keras and Tensor flow will be used for practical implementation and testing of the various methods taught in the course. Student will be given term paper to identify and solve a societal problem using the methods studied in the course.

FUTM-CPT 515: **Advanced Database Systems** (3 Units; C; LH-30, PH-45)

Learning Outcomes

Upon completion of this course, students are expected to:

i. know how relational databases are built using MySQL.

ii. evaluate the fundamental theories for advanced database architectures and query operators.

iii. state how metadata and its various tools are used in data warehousing.

iv. state and understand how large database systems are built from the existing databases as operational data sources.

v. describe the processes and building blocks of developing data warehouse in the area of business intelligence and deploy the knowledge for a practical hands-on.

vi. formulate metrics for data quality

vii. describe the aspects of big data analytics, algorithms involved and the challenges.

Course contents

Relational Data Models: Relational Constraints and Relational Algebra. Structured Query Language Relational Database Standard. Case Studies: Mysql. Data Warehousing: Introduction. What is Data Warehousing. Data Warehousing Concepts. Methodology for Data Warehousing. Issues in Data Warehousing. Benefits of Data Warehousing. Data Warehousing Building Blocks: Defining Features. Data Warehouse and Data Mart. Overview of Components. Matadata:

Abstraction. Use of Metadata in Data Warehousing. Tools for Metadata. Data Design and Data Preparation: ETL Overview. Data Extraction. Data Transformation. And Data Loading. Data Quality: Why is Data Quality Critical. Challenges. Tools OLAP In The Data Warehousing: Demand for OLAP – Major Features and Functions (Drill-Down. Rollup. Slice And Dice). OLAP Models – OLAP Tools: Web OLAP Approaches. OLAP Engine Design.

FUTM-CPT 521: **Big Data Analytics** (3 Units; C; LH-30, PH-45)

Learning Outcomes

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

i. state at least three the big data characteristics and challenges
ii. identify and use the existing big data processing platforms/tools
iii. outline the process of big data collection, integration and storage
iv. list at least 5 properties of the MapReduce Big data processing paradigms
v. implement various queries over big data
vi. identify the core techniques of analytics on big data
vii. list at least 4 different real applications and their techniques that involve big data

Course Contents

Overview of Big Data. State-of-the-art big data computing paradigms (Map-Reduce). Big data programming tools (e.g., Hadoop, MongoDB, Spark, etc.). Big data extraction and integration. Big data storage. Scalable big data indexing. Large-scale graph processing techniques. Big data stream techniques and algorithms. Large-scale probabilistic data analysis. Big Data Analytics with R. Big data privacy. Big data visualizations. Problems in real applications of big spatial-temporal data (e.g., geographical databases) Problems in real applications of big financial data (e.g., time-series data). Problems in real applications of big multimedia data (e.g., audios/videos). Problems in real applications of big medical/health data. Problems in real applications of big social media data. Problems in real applications of big scientific data (e.g., bioinformatics data).

Practical: Practice on cluster computing programming based on Map-Reduce pattern such as Hadoop MapReduce, Apache Spark and Flink. Analysing Twitter Data using Spark and MongoDB.Build machine learning dashboards using R and R Shiny, create web-based apps using NoSQL databases.

FUTM-CPT 522: **Computer System Performance Evaluation** (3 Units; C; LH-45,)

Learning Outcomes

On completion of the course, students should be able to:

i. state three goals of Performance Evaluation,
ii. describe two performance evaluation techniques
iii. compare Common Processor and System performance Metrics
iv. outline two types of benchmarks
v. differentiate between Benchmark Strategies and Benchmark Programs,

vi. state two types of workloads

vii. compare performance over a benchmark suite using the most appropriate type of average

Course Contents

Basic Concepts of system performance evaluation. Goals of Performance Evaluation. Techniques of System performance evaluation. Metrics of Performance. Attributes of Good Performance Metrics. Common Processor and System performance Metrics. Speedup and Relative Change. Measurement Techniques. Measurement strategies. Monitors. Statistics for Performance Analysis. Basic Probability and Statistics Concepts. Types of averages and Quantifying Variability. Benchmarking. Types of Benchmarks and Benchmark Strategies, Examples of Benchmark Programs. Types of Workloads and Workload Selection. Aggregating performance metrics over a benchmark suite. Aggregating Ratio Metrics and Aggregating Normalized Values. Statistical Sampling for Processor and Cache Simulation. Sampling for caches. Trace sampling for processors.

FUTM-CPT 524: **Introduction to Computer Vision** (3 Units; C; LH-30; PH-45)

Learning Outcomes

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

list and explain the fundamentals of computer vision, including image formation, feature extraction, and object recognition.

implement and use computer vision algorithms and techniques, including deep learning and convolutional neural networks.

list and use at least 2 computer vision libraries and tools, such as OpenCV and TensorFlow.

apply computer vision techniques to real-world problems, such as image and video analysis, object detection and tracking, and scene understanding.

develop understanding of the limitations and challenges of computer vision, such as dealing with variability in illumination, scale, and viewpoint.

execute further research and development in the field of computer vision and related areas.

apply critical thinking and problem-solving skills from computer vision concepts to solve various problems.

Course Content

Introduction. Image Formation. Images without lenses. The pinhole camera. Lens systems. Scaled orthographic projection. Light and shading. Color. Simple Image Features. Edges. Texture. Optical flow. Segmentation of natural images. Classifying Images. Modern computer vision methods such as Convolutional Neural Networks, deep learning methods for handling images and videos. Object Detection. Localisation and Recognition. Object tracking and motion estimation. Deploying computer-vision solutions for real world problems. Image classification with convolutional neural networks. The 3D World. 3D cues from multiple views. Binocular stereopsis.3D cues from a moving camera. 3D cues from one view. Using Understanding what people are doing. Linking pictures and words. Reconstruction from many views. Geometry from a single view. Making pictures. Controlling movement with vision.

Practical:

Students will perform various image processing techniques, such as image filtering, edge detection, and morphological operations, using popular computer vision libraries such as Open CV. Object Detection and Tracking: Students will implement object detection and tracking algorithms using Haar cascades, HOG, and deep learning-based object detection. Image Classification: Students will implement image classification algorithms using k-nearest neighbors, decision trees, and deep learning-based image classification. Image Stitching and Panorama: Students will implement image stitching and panorama algorithms using homography and RANSAC. Face Detection and Recognition: Students will implement face detection and recognition algorithms using Viola-Jones, HOG, and deep learning-based face recognition. Image Segmentation: Students will implement image segmentation algorithms using thresholding, region-based segmentation, and active contours. Feature Detection and Extraction: Students will implement feature detection and extraction algorithms using corner detection, Harris corner detection, SIFT, and SURF.

CSC 597: **Final Year Project I** (3 Units C: PH 135)

Learning Outcomes

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

i. identify a researchable project topic in Computer Science;
ii. search and review literature pertinent to identified problem statement;
iii. acknowledge and reference sources of information used in the research report;
iv. conceptualise and design a research methodology to address an identified problem;
v. determine tools for analysing data collected based on research objectives;
vi. write a coherent proposal on the research project to be conducted; and
vii. orally present the written project proposal.

Course Contents

An independent or group investigation of appropriate software, hardware, communication and networks or IT related problems in Computer Science 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 oralpresentation may also be required.

CSC 598: **Final Year Project II** (3 Units C: PH 135)

Learning Outcomes

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

i. demonstrate technical skills in Computer Science;
ii. demonstrate generic transferable skills such as communication and team work;
iii. produce a technical report in the chosen project;

iv. defend the written project report; and

v. appreciate the art of carrying out full-fledged research.

Course Contents

This is a continuation of CSC 597. This contains the implementation and the evaluation of the project. A formal written report, chapters 4-5 have to be approved by the supervisor. A final report comprising chapters 1 - 5 will be submitted to the department for final grading. An oralpresentation is required.

DISTRIBUTION OF MARKS

ITEM	MARKS	HOW TO CONDUCT EXAM
Project report	50	Assessed jointly by Internal and External examiners.
Oral Presentation	50	Assessed jointly by Internal and External examiners.

INDUSTRIAL TRAINING (SIWES)

At the end of first semester, 400 level students in the programme are expected to proceed on twenty-four (24) weeks of students' industrial work experience scheme (SIWES), which is to be done in an establishment that can provide relevant industrial experience to the students.

CHAPTER FOUR

FACILITIES AVAILABLE:

University Library: The University library has large collections of required, current and quality books, journals and other relevant literature for students and the department.

Departmental Library: Apart from the resources available at the main library, the department also operates its own Library with specialized collection to cater for the research and teaching needs of our lecturers and students. This is presently located in the school's faculty library in the School of Information and Communication Technology (SICT) complex. In addition to this, there is also an e-library with current and up to date books in the field of computer science and allied fields. There is provision for Local Area Network with Internet connectivity.

Computer Laboratory: The Department has two (2) standard, well equipped and functional computer laboratories for students' practical as well as for teaching. Apart from this, there are other general laboratories for the faculty with necessary and adequate software and hardware.

STAFFING

The Department has human resources group under academic staff (core staff on the ground for the programme and staff available for the programme from other sources such as Engineering and Science); non-academic staff (made up of secretary, clerical staff, and cleaners), and technical staff (system analyst, programmer, technicians, and technologist).

ACADEMIC STAFF LIST

S/N	Name of Staff	Area of Specialization	Qualification	Rank /Date of Appointment	Employment Status	
1.	AMINU, Enesi Femi	Artificial Intelligence: Knowledge Representations (Ontology Design), Machine Learning, and Smart Agriculture	Ph.D Computer Science (2023), M.Sc.Computer Science (2014) B. Sc. Computer Science (2008)	Senior Lecturer	TENURE	Head of Department
2.	ALHASSAN, John Kolo	Human Computer Interaction, Software Engineering, Artificial Intelligence.	Ph.D. Computer Science, (2014), M.Sc. Computer Science, (2006) B.Tech. Maths/Computer Science (2000)	Professor	TENURE	Director Academic Planning
3.	ABISOYE, Opeyemi Aderiike	Artificial Intelligence, Bioinformatics, High Performance Computing.	Ph.D Computer Science (2018), M.Sc.Computer Science (2010) B. Sc. Computer Science (2004)	Ass. Professor	TENURE	Sabbatical
4.	ADEPOJU, Solomon Adelowo	Human Computer Interaction, Web Mining, ICT 4D	PhD. Computer Science (2021) M.Sc. Computer Science (2006) B.Tech. Maths/Computer Science (2002)	Ass. Professor	TENURE	Lecturer

5.	OJERINDE Oluwaseun A.	Telecommunication Systems, Software Engineering, Health Care System & Blockchain Technology	Ph.D Electrical Engineering – Mobile Communication System(2014) M.Sc. Mobile Communication System(2008), Postgraduate Diploma in Electronic and Electrical Engineering(2018) B.Sc(Hons) Computer Science(Technology)(2006),	Ass. Professor	TENURE	HoD Software Engineering
6.	BASHIR, Sulaimon Adebayo	Machine Learning, Mobile Data Learning	PhD Computer Science (2017) M.Sc. Computer Science (2008) B.Tech. Computer Science (2003)	Senior Lecturer	TENURE	Deputy Dean
7.	MOHAMMED, Danlami Abdulmalik	Image Processing, Computer Vision & Machine Learning	Ph.D. Computer Science (2017) M.Sc. Computer Science (2010) B.Sc. Computer Science (2003)	Senior Lecturer	TENURE	Leave of Absence
8.	MUHAMMAD, Kudu Muhammad	Privacy Preservation of Learners, Data in Mobile Learning System Environment using Permission Blockchain Technology	Ph.D. Computer Science (2024) M.Sc. Computer Science (2014) B.Sc. Computer Science (2000)	Lecturer I	TENURE	Leave of Absence

9.	UGWOKE, Cosmas Uchenna	Information System & Mobile Computing	M.Sc. Computer Science (2010) B.Tech. Maths/Computer Science (2007)	Lecturer I	TENURE	On Study leave
10.	MOHAMMED, Idris Kolo	Biometric Intrusion Detection System	M.Tech Computer Science (2015) B.Tech. Mathematics/ Computer Science (2007)	Lecturer I	TENURE	Turnitin officer
11.	ADAMA, Victor Ndako	Human Computer Interaction, Usable Security	M.Tech. Computer Science (2016) B.Tech. Computer Science (2008)	Lecturer I	TENURE	On Study Leave
12.	EKUNDAYO, Ayobami	Data Mining	M.Sc. Computer Science (2016) B.Sc. Computer Science (2012)	Lecturer I	TENURE	100L Adviser
13.	LASOTTE Yakubu Boyi-Musa	Distributed Computing	M.Tech. Computer Science (2023) B.Tech Computer Science (2015)	Assistant Lecturer	TENURE	Project Co-ordinator
14.	SANI Alkali Umar	Machine Learning	B.Sc. 1^1 (2018)	Graduate Assistant	TENURE	On Study Leave
15.	LAWAL Olamilekan Lawal	Distributed Computing	B.Sc. 2^1 (2013)	Assistant Lecturer	TENURE	300L Adviser

16.	SAIDU Ahmed Abubakar	Distributed Computing	B.Tech 2^1 (2021) M.Tech (In View) (2022)	Graduate Assistant	TENURE	200L Adviser
17.	ABDULLAHI Dogara Kabiru	Distributed Computing	BSc. 2^1 (2023) M.Tech (In View)	Graduate Assistant	TENURE	Lecturer
18.	SHUAIBU Muhammad Badeggi	Distributed Computing	B.Tech 2^1 (2015) M.Tech (In View)	Graduate Assistant	TENURE	On Study Leave
19.	BASHIR, Kasim Muhammad	Distributed Computing	B.Sc 2^1 (2021) M.Tech (In View) 2024	Graduate Assistant	TENURE	On Study Leave
20.	ABDULKADIR, Mubarak Sharu	Distributed Computing	B.Sc 2^1 (2018) M.Tech (In View)	Graduate Assistant	Permanent and Pensionable	On Study Leave
21.	ABDULLAHI, Anas	Distributed Computing	B.Sc 2^1 (2021) M.Tech (In View)	Graduate Assistant	Permanent and Pensionable	On Study Leave

NON-ACADEMIC STAFF

S/N	Name	Rank/	Qualifications
1	ABUBAKAR, Yahaya	Deputy Director System Analysis	BTech. Math/ Computer Science, (2000) MSc.2010
2	OSAGIE, Cynthia Ibhande	Assistant Chief Technologist	ND Computer Science, (2005). HND Computer Science. (2008). PGD Computer Science. (2019- 2023)
3	MUHAMMAD, Isah Baba	Technologist I	HND Computer Science (2019)
4	ADAMU MUSA	Technologist II	HND Computer Science (2013)
5	ANAGO, Emily Ojone	Confidential Secretary	N.C.E, 2009, WASSCE, 2009
6	GBATE, Ibrahim Umaru	Higher Executive Officer (HEO)	National Diploma Public Administration (2013)