Department of Compute Science

Faculty of Applied Science

Trincomalee Campus, Eastern University, Sri Lanka

Student Handbook

Academic Year 2022/2023

Department of Computer Science

Faculty of Applied Science

Trincomalee Campus, Eastern University, Sri Lanka

Vision

Formation of Highly Intellectual Capital to the Computer world.

Mission

Department of Computer Science of the Faculty of Applied Science, Trincomalee Campus aims to provide highly marketable dynamic Computer technological graduates through ~~the~~ suitable teaching and learning environments.

MESSAGE FROM THE DEAN/ FACULTY OF APPLIED SCIENCE

As the Dean of the Faculty of Applied Science, it gives me great pleasure to welcome all of you to the Department of Computer Science, Faculty of Applied Science, Trincomalee Campus. First and foremost, congratulations to all of you. This is the beginning of your journey as a first-year Computer Science student. You are the cream of the crop among the students who sat for the Advanced level examination. Your hard work and dedication have brought you to this point, and you should be immensely proud of your accomplishments. You are now part of a community that values knowledge, critical thinking, and innovation. Embrace this privilege and let it inspire you to push boundaries, challenge assumptions, and make a meaningful impact in the world. I assure you that the years ahead will be filled with remarkable opportunities and countless moments that will shape your lives forever.

The world of Computer Science is vast and constantly evolving. The demand for computer science professionals continues to grow across various industries. Almost every sector, including finance, healthcare, entertainment, and manufacturing, relies on computer systems and software. Pursuing a computer science degree equips you with the skills and knowledge needed to meet this demand and opens up a wide range of career opportunities. Engaging in research projects and internships will build a strong network within the industry. Your time at this Campus will provide you with numerous opportunities to explore, experiment, and excel.

The campus is not merely about acquiring knowledge from textbooks or sitting through lectures. Your education extends far beyond the walls of this institution. Make connections, join clubs and organizations, and participate in community service involved in sports and cultural activities. Though you came from various parts of Sri Lanka, now you all are under one roof. There are Tamils, Singhalese and Muslims among you. By actively engaging with different religious and cultural groups, you can contribute to the promotion of social harmony and ethnic cohesion.

During your stay in the Campus, in addition to the academic qualification try to improve communication skills and teamwork ability. This is a golden opportunity to learn the Tamil language from Sinhala students and the Sinhala language from Tamil students. Communication skills and teamwork ability are highly valued in the job market and can significantly enhance your chances of finding better job opportunities. Most of our graduates got jobs in government and private sectors immediately after graduation.

You have to strictly follow the Campus rules and regulations, otherwise, you will face disciplinary action by the Campus administration. I wish you all the best for your foreseeable future.

MESSAGE FROM THE HEAD/ DEPARTMENT OF COMPUTER SCIENCE

As the Head of the Department of Computer Science, Faculty of Applied Science, Trincomalee Campus, Eastern University, Sri Lanka, I am very happy to say few words regarding the current trend and the opportunities in the field of computer science.

The specialty of computer sciences keeps making continuous changes in the different aspects of life. The world has witnessed and keeps witnessing technological waves whose basis is research and applications of computer science; all of which has directly affected our daily lives. In recent years, many international universities have witnessed a noticeable increase in the rate of students' demand on the specialty of computer science. This is only because young people are aware of the importance of this specialty, the employment opportunities it provides and its impact on society.

With your Computer Science knowledge, you can, for example, create mobile sites and applications, analyse data and develop information, manage databases efficiently, ensure data integrity and confidentiality, and even study diseases and discover their relationship to drugs. All of this could happen while you're eager to discover and learn a solution.

Remember! Research is the turning point for all these.

STAFF OF THE DEPARTMENT OF COMPUTER SCIENCE

ACADEMIC STAFF

Head/DCS
Mr. S. Thadchanamoorthy
Senior Lecturer Gr-I
M.Phil. (CS) [UoC]
M.Sc. (CS) [UoC]
***B. Sc. (Elect. & Electronics Eng)* [UoP]**

Ms. K. Krishnaraj
Lecturer
M.Sc. (CS) [SAU, India]
B.Sc. Hons (ICT) [VCUJ]

Ms. Y. Kalyani
Lecturer (Probationary)
B.Sc. (CS) [UoJ]
PhD Reading..

Ms. T. Thanushya
Lecturer
M.Phil. (CS) [UoK]
M.Sc. (CS) [UoP]
B.Sc. Spl (CS) [VCUJ]

Mr. A. Suthakaran
Lecturer (Probationary)
M.Sc. (IT) [UoM]
***B.Sc. (CS) [UoJ*]**

Ms. K. Tharmini
Lecturer (Probationary)
B.Sc. Hons (CS) [UoJ]

STAFF OF THE DEPARTMENT OF COMPUTER SCIENCE

ACADEMIC STAFF

Ms. K. Disne
Lecturer (Probationary)
M.Sc. (IT) [UoM]
B.Sc. Hons (CS) [EUSL]

Ms. S. Priyanka
Lecturer (Probationary)
M.Sc. (CS) [UoM]
B.Sc. Spl (CS) [SEUSL]

Ms. J. Janani
Lecturer (Probationary)
B.Sc. Hons (CS) [UoJ]

Ms. P. R. Vithusia
Lecturer (Probationary)
M. (CS) [UoP]
B.Sc. Spl (CS & Tech) [UWU]

ACADEMIC SUPPORTIVE STAFF

Mr. W. Sriwathsan
Instructor in Computer Technology Gr I
M.Sc. (CS) [UCSC]
B.Sc. (Phys Sc) [EUSL]

Mr. Benjamin Christopaul
Instructor in Computer Technology Gr II
M.Sc. (Sc Edu) [EUSL]
M.Sc. (CS) [UoP]
B.Sc. (Phys Sc) [EUSL]

NON-ACADEMIC STAFF

Mrs. U. Telahini
Management Assistant

Mr. N. Sivakumar
Works Aide

1. COMPUTER SCIENCE DEGREE PROGRAMME

The three year Computer Science Degree is named as follows:

- ✓ Bachelor of Science in Computer Science – BSc (CS)

1.1. INTRODUCTION

The Faculty of Applied Science conducts three-year degree programmes under the semester-based, course system in English medium. Each semester consists of 15 weeks of academic activities. Each academic year will be considered as level 1, level 2 and level 3 respectively.

The programme is entitled as Bachelor of Science in Computer Science [BSc (CS)]. Initially (from 2007/2008 academic batch) the degree programme was named as Bachelor of Computer Science which is also equivalent to the SLQF level 5. But the name of the degree programme is mentioned as Bachelor of Science in Computer Science in the UGC Admission Handbooks. The Senate of Eastern University, Sri Lanka, at its 343rd meeting held on 17.05.2023 recommended the name change, as Bachelor of Science in Computer Science, as per the SLQF standard.

1.2. AIM OF THE PROGRAMME

Bachelor of Science in Computer Science degree programme aims to prepare graduates to succeed in a rapidly changing field. This will support graduates for professional careers, lifelong learning and serving the community in a professional manner.

1.3. GRADUATE PROFILE AND ATTRIBUTES

Computer Science graduates will possess the ability to integrate theory and practice, recognize the importance of abstraction, and appreciate the value of good computer engineering design. At the broad level, BSc (CS) graduates will possess the following set of attributes:

- **Knowledgeable in computer science**: Graduates demonstrate knowledge and understanding of essential facts, concepts, principles, and theories relating to computer science and software applications.
- **Problem solver**: Graduates need to understand how to apply the appropriate knowledge and skills, including background research and experimentation, to identify,

investigate, abstract, conceptualize, analyse, and solve complex computing problems, in order to reach substantiated conclusions.

- **Significant project experience**: Project demonstrates the practical application of principles learned in different courses and forces students to integrate material learned at different stages of the curriculum. Students need to appreciate the need for domain knowledge for certain applications, and that this may necessitate study within that domain.
- **Lifelong learner**: Graduates should learn new tools, computer languages, technologies, techniques, standards and practices, as well as be able to identify and address their own educational needs in a changing world in ways sufficient to maintain their competence and to allow them to contribute to the advancement of knowledge.
- **Act Professionally**: Graduates should act appropriately with respect to ethical, societal, environmental, health, safety, legal, and cultural issues within local and global contexts, and with regard to the consequential responsibilities relevant to professional computing practice.
- **Effective communicator**: Graduates should be able to communicate with the computing community and with society at large about complex computing activities by being able to comprehend and write effective reports, design documentation, make effective presentations, and give and understand clear instructions.

1.4. PROGRAMME INTENDED LEARNING OUTCOMES

The following expected student's outcomes apply to the Computer Science degree programme. Students graduating from the Bachelor of Science in Computer Science programme will be,

- Able to use a range of programming languages and tools to develop computer programs and systems that are effective solutions to problems.
- Able to understand, design, and analyse precise specifications of algorithms, procedures, and interaction behaviour.

- Able to apply mathematics, logic, and statistics to the design, development, and analysis of software systems.
- Able to be equipped with a range of fundamental principles of Computer Science that will provide the basis for future learning and enable them to adapt to the constant rapid development of the field.
- Able to have experience working in teams to build software systems.

1.5. VOLUME OF LEARNING

1.5.1 COURSE CREDITS

The volume of learning is described in terms of credits. One credit is equivalent to 50 notional learning hours. The notional learning hours include direct contact hours with teachers and trainers, time spent in self- learning, preparation for assignments, carrying out assignments and assessments.

- ✓ One credit of the taught course, laboratory studies is equivalent to 50 notional learning hours.
  - 15 hours of lectures and 35 hours of independent learning and assessments; or
  - 30 hours of laboratory work with additional time for independent learning and assessments; or
- ✓ One credit of industrial training or research project (including time allocated for literature survey) is considered equivalent to a minimum of 100 notional hours.

Credits have to be earned by students after successful completion of the work required and appropriate assessment of learning outcomes.

1.5.2 CREDITS OF THE DEGREE PROGRAMME

The Degree programme is offered in six semesters. Each student should earn a minimum of 90 credits to complete the degree.

1.5.3 Maximum Duration of the Degree Programme

All students should complete their degrees within a specified period of time. The maximum period allowed for the Degree programme will be six (6) academic years from the date of first registration.

1.6. INDUSTRIAL TRAINING/PROJECT

All students must undergo industrial training at the end of the sixth semester. Therefore, soon after the sixth semester examination, students should be able find suitable IT industry / Software Company for a period of six months, which carries 3 credits GPA.

2. COURSE STRUCTURE

2.1 COURSE CODE

2.1.1 CORE COURSES

Every course is assigned a course code. The code will be of the pattern XX YSCN, where;

First two characters (XX) refer to

if

CO	*then*	**Co**mmon Core course
CS	*then*	**C**omputer **S**cience course
EC	*then*	**E**lective **C**ourse

Third character (**Y**) refers to **Y**ear

Fourth character (**S**) refers to **S**emester

Fifth character (**C**) refer to **C**redit

Last character (**N**) refers to Subject **n**umber

2.1.2 AUXILIARY COURSES/ ELECTIVE COURSES

The Auxiliary courses/Elective courses are compulsory, but not taken for the computation of Grade Point Average (GPA); however shall be evaluated and appear in academic transcript, and be partial requisite for the award of degree. The students are expected to obtain at least C pass in these course examination prior to the award of degree.

General English Proficiency	- Auxiliary course
Research Work	- Auxiliary course
Foundations of Management	- Elective course

2.2 OUTLINE OF THE COURSE STRUCTURE

The Degree is named as follows;

Bachelor of Science in Computer Science - 3 Years

The following courses are offered in the above degree programme.

Where: **L** denotes **Lecture** hours, **P** denotes the **P**ractical hours and **IL** denotes **I**ndependent **L**earning hours, based on the notional hours described in the SLQF standard.

Year I: Semester I

Course Code	Course Title	Hours	Credit
		L/P/IL	
CO1121	Basic Mathematics for Computing	30/00/70	2
CO1122	Basic Computer Programming	30/00/70	2
CO1112	Practical work on CO1122	00/30/20	1
CO1123	Formal Methods for Problem Solving	30/00/70	2
CO1124	Computer Systems & PC Applications	30/00/70	2
CO1114	Practical work on CO1124	00/30/20	1
CO1125	Statistics for Science and Technology	30/00/70	2
CO1115	Practical work on CO1125	00/30/20	1
CO1126	Management Information System	30/00/70	2
GEP - I	General English Proficiency - I	30/00/70	-
			15

Year I: Semester II

Course Code	Course Title	Hours	Credit
		L/P/IL	
CO1221	Systems Analysis & Design	30/00/70	2
CO1222	Data Structures & Algorithms	30/00/70	2
CO1212	Practical work on CO1222	00/30/20	1
CO1223	Data Base Management Systems	30/00/70	2
CO1213	Practical work on CO1223	00/30/20	1
CO1224	Multimedia & Hypermedia Development	30/00/70	2
CO1214	Practical work on CO1224	00/30/20	1
CO1225	Computer Architecture	30/00/70	2
CO1226	Social Harmony	30/00/70	2
			15

Year II: Semester I

Course Code	Course Title	Hours	Credit
		L/P/IL	
CO2121	Advanced Mathematics for Computing	30/00/70	2
CO2122	Operating Systems	30/00/70	2
CO2112	Practical work on CO2122	00/30/20	1
CO2123	Software Engineering	30/00/70	2
CO2124	Internet and Web Design	30/00/70	2
CO2114	Practical work on CO2124	00/30/20	1
CO2125	Object Oriented Programming	30/00/70	2
CO2115	Practical work on CO2125	00/30/20	1
CO2126	Sri Lankan Studies	30/00/70	2
GEP - III	General English Proficiency - III	30/00/70	-
			15

Year II: Semester II

Course Code	Course Title	Hours	Credit
		L/P/IL	
CO2221	Data Communication Systems	30/00/70	2
CO2222	Visual System Development Tools	30/00/70	2
CO2212	Practical work on CO2222	00/30/20	1
CO2223	Computer Graphics	30/00/70	2
CO2213	Practical work on CO2223	00/30/20	1
CO2224	Human Computer Interaction	30/00/70	2
CO2214	Practical work on CO2224	00/30/20	1
CO2225	Software Management Techniques	30/00/70	2
CO2226	Automata Theory	30/00/70	2
			15

Year III: Semester I

Course Code	Course Title	Hours	Credit
		L/P/IL	
CS3121	Logic Programming & Expert Systems	30/00/70	2
CS3111	Practical work on CS3121	30/00/70	1
CS3122	Advanced Database Management Systems	30/00/70	2
CS3112	Practical work on CS3122	00/30/20	1
CS3123	Systems & Network Administration	30/00/70	2
CS3113	Practical work on CS3123	00/30/20	1
CS3124	Data Security	30/00/70	2
CS3114	Practical work on CS3124	00/30/20	1
CS3135	Theory of Computing	30/00/70	3
	Elective Course (1 of 3)		
EC3101	Foundations of Management	45/00/105	-
EC3102	Organizational Behaviour	45/00/105	-
EC3103	Financial Accounting	45/00/105	-
			15

Year III: Semester II

Course Code	Course Title	Hours	Credit
		L/P/IL	
CS3221	Assembly Programming	30/00/70	2
CS3211	Practical work on CS3221	00/30/20	1
CS3222	Software Quality Assurance	30/00/70	2
CS3212	Practical work on CS3222	00/30/20	1
CS3233	Professional Issues in IT	45/00/105	3
CS3224	Computer Networks	30/00/70	2
CS3214	Practical work on CS3222	00/30/20	1
CS3235	Industrial Training/Project	00/00/300	3
	Research Work*	00/00/300	-
			15

***** The students who want to obtain a BSc (CS) degree, should complete a Research Work of 3 Non-GPA credits during the sixth semester (Year III Semester II).

3. EXAMINATION STRUCTURE

Examinations are conducted at the end of each semester followed by a study leave. Study leave is given for a period of two (2) weeks at the end of each semester. The semester examination is conducted within a period of four (4) weeks. The duration of final theory and practical examination may vary according to the credit value of the course, as follows:

Theory

Credit	Exam Duration	Number of Questions
1	1 hour	2
2	2 hours	4
3	3 hours	5 or 6

Practical

Credit	Exam Duration	Number of Questions
1	2 hours	2
> 1	3 hours	3 or 4

Allocated percentage marks for sub-questions of each question of the summative examination shall be specified (denoted) in the question paper.

3.1 ATTENDANCE

All registered students are required to attend all lectures, tutorials, and practical classes. 80% attendance is compulsory for both theory and practical in each course. Any student who does not achieve 80% attendance will not be allowed to sit for the semester examination of that course.

4. EVALUATION SYSTEMS

4.1 INTRODUCTION

Evaluation consists of Formative (continuous) and Summative (end semester) assessments.

Formative assessment usually accounts for 35% of the total marks. Formative assessments may consist of mid-semester examinations, assignments, quizzes given in class, take-home assignments such as papers or problem sets, in-class presentations by students, projects, etc.

All continuous assessments conducted shall be taken into computation, but the weight for different types of assessment (i.e. not equal weight for assignments, quizzes, etc.) shall be decided by the lecturer in charge and expected to be announced to the students at the beginning of the course.

Theory

The final mark (M_1) for the theory examination in a course unit will be evaluated using the following equation:

$$M_1 = T * 0.65 + A_T * 0.35$$

Where T is the marks obtained in the final theory examination and A_T is the marks obtained in continuous assessment during the course.

Practical

The final mark (M_2) for the practical course will be evaluated as follows:

$$M_2 = P * 0.65 + A_P * 0.35$$

Where P is the marks obtained in the final practical examination and A_P is the marks obtained in the continuous assessment.

Industrial Training/Project

Industrial training after the 6th semester will be assessed and the marks will be allocated as follows:

Components	Marks
Final Viva- Voce Examination	40%
Final Report	60%
Total	**100%**

Both the report and viva-voce examination are mandatory. Students should obtain a minimum of 50 % in each component. Failing in any of these components will be considered to repeat the work, according to the time frame given by the Head of the Department.

Research Work

Research work during the 6th semester will be assessed at the end of the written examination and the marks will be allocated as follows:

Components	Marks
Final Presentation	20%
Final Viva- Voce Examination	20%
Research Report	60%
Total	**100%**

Research report, presentation and viva-voce examinations are mandatory. Students should obtain a minimum of 50 % in each component. Failing in any of these components will be considered to repeat the work, according to the time frame given by the Head of the Department.

4.2 GRADING SYSTEM AND GRADE POINT AVERAGE

Based on the scheme of evaluation mentioned above, marks obtained in respect of a course unit will be graded based on UGC Commission circular: 901.

Marks % Range	Grade	Grade Point Value
75 – 100	A+	4.00
70 – 74	A	4.00
65 – 69	A-	3.70
60 – 64	B+	3.30
55 – 59	B	3.00
50 – 54	B-	2.70
45 – 49	C+	2.30
40 – 44	C	2.00
35 – 39	C^-	1.70
30 – 34	D^+	1.30
25 – 29	D	1.00
00 - 24	E	0.00

Grade Point Average (GPA) is the credit-weighted arithmetic mean of the Grade Point Value which is formulated as

$$GPA = \frac{Sum\ of\ (credits \times grade\ point\ value)}{Total\ credits} = \frac{\sum c_i g_i}{\sum c_i}$$

Where c_i is the number of credits for the i^{th} course and g_i is the grade point value for the i^{th} course. The GPA is calculated for each academic year.

The Overall GPA (OGPA) for the degree programme would be the credit weighted average will be equivalent to

$$\text{OGPA} = \frac{G_1+G_2+2G_3}{4}$$

Where, $G_{1,}$ $G_{2,}$ G_3 are the GPA for the first, second and third year of study respectively.

4.3 REPEATING COURSES

- Those who fail to obtain the requisite number of credit passes or fail to appear for an end-semester examination are required to appear for such an examination when it is held next.
- A student who obtains less than a grade C for a course shall be repeated.
- Only end-semester examination marks shall contribute to the final grade for a repeating course.
- Maximum grade for the repeat examination will be ***C***.
- Repeat examination of a candidate supported by a medical certificate either by the campus medical officer (CMO) or certified by the CMO will be considered as his/her first attempt. Such Medical Certificate should be submitted with the appropriate certification of CMO within two (2) weeks from the date of the said examination held.
- The special repeat examination for the final year students may be conducted within 45 days from the release of third-year second-semester results.
- Examination for a course can be repeated not more than three times. A grace chance is permitted with the approval of the Senate of the EUSL.
- Candidates will not be permitted to re-sit any passed course, but will be given the option to repeat a course with a ***C-*** or lower grade to improve it.
- In the event a candidate obtains a lower grade while attempting to improve the grade, he or she will be entitled to the previous grade.

5. AWARDS

5.1 AWARD OF BACHELOR OF SCIENCE IN COMPUTER SCIENCE

To be eligible for Bachelor of Science in Computer Science Degree Programme, a candidate must obtain

1. C or better grades for at least 72 credits and C- grades for the remaining 18 credits, of which not more than 6 from each year of study; and
2. a minimum overall GPA of 2.0 in first, second and third years of study; and
3. a minimum of C grade from each of the General English courses; and
4. a minimum of C grade from the course Research Work; and
5. the relevant requirements within a period of six academic years.

5.2 AWARD OF CLASSES

A candidate who has fulfilled all the conditions for the award of BSc (CS) degree shall be awarded a class, if he/she fulfils the following additional requirements:

First Class:

1. Obtain a minimum OGPA of 3.70; and
2. Obtain A or better grades in at least 36 credits, of which at least 12 credits from the third year of study; and
3. Complete the degree programme within the three academic years.

Second Class (Upper Division)

1. Obtain a minimum OGPA of 3.30; and
2. Obtain B or better grades in at least 36 credits of which at least 12 credits from the third year of study; and
3. Complete the degree programme within the three academic years.

Second Class (Lower Division)

1. Obtain a minimum OGPA of 3.00; and
2. Obtain B or better grades in at least 36 credits of which at least 12 credits from the third year of study; and
3. Complete the degree programme within the three academic years.

5.3 EFFECTIVE DATE OF THE DEGREE

The effective date of the degree shall be the last date of the final year written paper examination or viva voce examination of Industrial Training whichever comes last.

5.4 FALLBACK QUALIFICATIONS

A fall back qualification may be awarded to an undergraduate student only after completion of the maximum period of study (6 Years) of the Bachelor of Science in Computer Science degree programme. However, if a student is expelled from the University on disciplinary grounds, then such a student may not be eligible for the award of fall back qualification.

5.4.1 AWARD OF DIPLOMA

A candidate who has failed to complete the requirements of the General degree within a period of six academic years but achieved some significant level will be awarded a ‘Diploma’ if s/he has

a) C or better grades for at least 24 credits and C- grades for the remaining 6 credits;

b) obtained a minimum OGPA of 2.00; and

c) Fulfil the relevant requirements for the Diploma mentioned above (a) and b)) within a period of six academic years.

5.4.2 AWARD OF HIGHER DIPLOMA

A candidate who has failed to complete the requirements of the General degree within a period of six academic years but achieved some significant level will be awarded a ‘Higher Diploma’ if s/he has

a) C or better grades for at least 48 credits and C- grades for the remaining 12 credits;

b) obtained a minimum OGPA of 2.00; and

c) Fulfill the relevant requirements for Higher Diploma mentioned above in a) and b) within a period of six academic years.

5.4.3 EFFECTIVE DATE OF FALLBACK QUALIFICATION

The effective date of award of a fall-back qualification will be the 1st day of the month after which the Senate has approved the award of the qualification, at the request of the student, and on the recommendation of the Faculty Board.

6. EXAMINATION RULES AND REGULATIONS

The following exam rules are in the Manual of Procedure on Conducting Examination (MPCE) published by the Academic Affairs Department of Eastern University, Sri Lanka in August 2022.

6.1 CANDIDATES ATTENDING THE EXAMINATION

This refers to the chapter XI of MPCE.

1	Candidates shall be in attendance outside the examination hall at least 15 minutes before the commencement of each paper, but shall not enter the halls until they are requested to do so by the supervisor.	Attendance
2	On admission to the hall, a candidate shall occupy the seat allotted to him/her and shall not change it except on the specific instructions of the supervisor. 4	Seating
3	No candidate shall be admitted to the examination hall for any reason whatsoever after the expiry of half an hour from the commencement of the examination. Nor shall a candidate be allowed to leave the hall until half an hour has lapsed from the commencement of the examination or during the last 15 minutes of the paper.	Admission to Exam hall
4	Candidates shall have their Student Record Book, Student Identity Card and Admission Card with them in the examination hall on every occasion they attend for a paper/ an exam. The candidature is liable to be cancelled if a student does not produce the Student Record Book. If a candidate fails to bring his/her record book on any occasion, he/she shall sign a declaration in respect of the paper for which he/she had not produced the record book in the form provided for it, and produce the record book on the next occasion when he/she appears for the examination. The presentation of the Record Book thus, should be documented on the declaration form. The declaration forms shall be checked by the DR/SAR/AR of the faculty before the release of results. If it is the last paper or the only paper he/she is sitting, they shall produce the record book to the DR/SAR/AR of the faculty on the following day, and get the documentation on the declaration form. If a candidate loses his/her record book in the course of the examination, he/she may present his/her Student Identity Card and shall obtain a duplicate record book from the DR/SAR/AR of the faculty, for producing at the examination hall.	Presenting Identification
5	No candidate shall have any notes, signs, formulae, mobile phones, smart watches, other communication devices or any other unauthorized documents on his person, in his clothes, on the admission card, time table or record book. Books, notes, parcels, hand bags, mobile phones, other information and communication devices etc. which a candidate has brought with him/her should be kept at a place indicated by the Supervisor/ Invigilator .	Documents etc. which candidates should not bring into the examination hall

6	A candidate may be required by the supervisor to declare any item in his possession or person.	Declaration of articles in possession
7	No candidate shall copy or attempt to copy from any book or paper or notes or similar material or from the scripts of another candidate. Nor shall any candidate either help another candidate or obtain help from another candidate or any other person. Nor shall any candidate conduct himself so negligently that an opportunity is given to any other candidate to read anything written by him/her or to watch any practical examination performed by him. Nor shall any candidate use any other unfair means or obtain or render improper assistance at the examination.	Candidates prohibited from copying/talking/exchange of answer books, use of mobile phones etc., violating Exam Offenses.
8	No candidate shall submit a practical or field book or dissertation or project study or answer script which has been done wholly or partly by anyone other than the candidate himself	Cheating or Plagiarism in the submission of work
9	Candidate shall bring their own pens, ink, mathematical instruments, erasers, pencils, or any other approved equipment or stationary which they have been instructed to bring.	Articles candidates may bring into Exam Halls
10	Examination stationery (i.e., writing paper, graph paper, drawing paper, ledger paper, précis paper etc.,) shall be supplied as and when necessary. No sheet of paper or answer book Examination stationery supplied to a candidate may be torn crumpled, folded or otherwise mutilated. No paper other than those supplied to him/her by the supervisor/ invigilator shall be used by candidates. Log tables or any other material provided shall be used with care and left behind on the desk. All the material supplied, whether used or unused, shall be left behind on the desk and not removed from the examination halls by the candidate.	Examination stationery university property
11	Every candidate shall enter his/her index number on the answer book and on every continuation paper. He/she shall also enter all necessary particulars as indicted in the cover of the answer book. A candidate who inserts on his script and index number other than his own is liable to be considered as having attempted to cheat. A script that bears no index number or an index number which cannot be identified, is liable to be rejected. No candidate shall write his name or any other identifying mark on the answer script.	Index Number
12	All calculations and rough work shall be done only on paper supplied for the examination and shall be cancelled and attached to the answer script. Such work should not be done on admission cards, time tables, question papers, record books or on any other paper. Any candidate who disregards these instructions runs the risk of being considered as having written notes or outline of answers with the intention of copying.	Rough work to be done on provided paper only and cancelled
13	Any answer or part of an answer which is not to be considered for the purpose of assessment shall be neatly crossed out. If the same question has been attempted in more than one place the answer or answers that are not to be counted shall be neatly crossed out	Unwanted parts of answers to be crossed out
14	Candidates are under the authority of the supervisor and shall assist him/her by carrying out his instructions and those of his	Under supervisors authority

	invigilators, during the examination and immediately before and after it.	
15	Every candidate shall conduct himself in the examination hall and its precincts so as not to cause disturbance or inconvenience to the supervisor or his staff or to other candidates. In entering and leaving the hall, he/she shall conduct himself as quietly as possible. A candidate is liable to be excluded from the examination hall for disorderly conduct.	Conduct
16	Candidates shall stop work promptly when ordered by the supervisor/ invigilator to do so.	Stopping work
17	Absolute silence shall be maintained in the examination hall and its precincts. A candidate is not permitted for any reason whatsoever to communicate or to have any dealings with any person other than the supervisor/ invigilator.	Maintenance of silence
18	During the course of answering a paper, no candidate shall be permitted to leave the examination hall temporarily. In case of an emergency, the supervisor/ invigilator shall grant him/her permission to do so but the candidate will be under his surveillance.	Leaving the Exam hall
19	No person shall impersonate a candidate at the examination, nor shall any candidate allow himself to be so impersonated by another person	Impersonation
20	Serious note will be taken of any dishonest assistance given to a candidate, by any person.	Dishonesty
21	If circumstances arise which in the opinion of the supervisor render the cancellation or postponement of the examination necessary, he/she shall stop the examination, collect the scripts already written and then report the matter as soon as possible to the Vice chancellor/ Registrar.	Cancellation/postponement
22	The supervisor/ invigilator is empowered to require any candidate to make a statement in writing on any matter which may have arisen during the course of the examination and such statement shall be signed by the candidate. No candidate shall refuse to make such a statement or to sign it.	Making of statement
23	No candidate shall contact any person other than the Vice Chancellor, Dean, Head of the Department or the Registrar regarding any matter concerning the examination.	Who to contact in exam. Matters.
24	Every candidate shall hand over the answer script personally to the supervisor/ invigilator or remain in his seat until it is collected. On no account shall a candidate hand over his answer script to the attendant, a minor employee or another candidate.	Handing over the answer script.
25	Every candidate who registers for an examination shall be deemed to have sat the examination unless he/she withdraws from the examination within the specified period or submits a medical certificate prior to the commencement of the examination. The medical certificate shall be from the university medical officer. If this is not possible the medical certificate should be obtained from	Withdrawal from Examination applied

	a Government Medical Practitioner, and submitted to the university medical officer at the earliest possible time.	
26	When a candidate is unable to present himself for any part/ section of an examination, he/she shall notify or cause to be notified this fact to the Registrar immediately. This should be confirmed in writing with supporting documents within 48 hours by registered post.	Absence from Exams
27	A student who withdraws or absents himself from the examination shall not be eligible for classes at the next examination unless the senate decides otherwise	Eligibility for Classes
28	No student shall sit an examination, if he/she has exhausted the number of attempts that he/she is allowed to sit that particular examination, unless he/she has been granted special permission to do so by the Senate	Eligibility to continue to sit an Exam, if number of attempts exhausted

7. PROCEDURES TO FOLLOW WHEN A CANDIDATE IS UNABLE TO ATTEND AN EXAMINATION

A candidate who has not appeared for an examination or part of it for which he/she is due to sit, should make an appeal if he/she wants to preserve the attempt for a future examination. This allowance shall not be considered unless the candidate makes an appeal.

The reasons may be considered under one of the following:

- ✓ Unexpected illness
- ✓ Death in immediate family and bereavement
- ✓ Other reasons that may be considered valid by the Faculty Board and Senate

7.1 UNEXPECTED ILLNESS

1	In case a candidate is unable to attend an examination or part of it due to illness, he/she should submit an appeal letter accompanied by a Medical Certificate (MC) issued by the doctor who has treated him/her to the Dean of the faculty within two weeks of the examination. The Dean shall send the MC to the University Medical Officer for authentication	Submission of Medical Certificate and deadline
2	Once the MC is authenticated the appeal shall be taken up at the Faculty Board and recommended to the senate through the DR/SAR/Academic Affairs, if found appropriate.	UMO approval of MC and Faculty Board recommendation
3	The senate shall then decide whether the appeal is acceptable and approve/deny the request. The decision shall be informed to the Faculty and the candidate by the Secretary of the Senate.	Acceptance of Appeal by Senate and information to candidate

7.2 DEATH IN IMMEDIATE FAMILY AND BEREAVEMENT

1	In case a candidate is unable to attend an examination or part of it owing to bereavement due to a death in his/her immediate family he/she should submit an appeal letter accompanied by evidence of such death. Immediate family here indicates one's own parents, siblings, spouse or children.	Submission of Appeal accompanies by evidence of death, of immediate family member
2	The appeal shall be taken up at the Faculty Board and recommended to the senate through the DR/SAR/AR of the faculty, if found appropriate.	Faculty Board recommendation
3	The senate shall then decide whether the appeal is acceptable and approve/deny the request. The decision shall be informed to the Faculty and the candidate by the Secretary of the Senate	Acceptance of Appeal by Senate and information to candidate

7.3 OTHER REASONS THAT MAY BE CONSIDERED VALID BY THE SENATE

1	In case a candidate is unable to attend an examination or part of it due to a reason which may be considered valid, the candidate shall submit an appeal letter accompanied by evidence of such. Examples: representing the Faculty, University or Country in any event approved by the VC/Dean	Submission of Appeal accompanies by evidence
2	The appeal shall be taken up at the Faculty Board and recommended to the senate through the DR/SAR/AR of the faculty, if found appropriate.	Faculty Board recommendation
3	The senate shall then decide whether the appeal is acceptable and approve/deny the request. The decision shall be informed to the Faculty and the candidate by the Secretary of the Senate	Acceptance of Appeal by Senate and information to candidate.

8. EXAMINATION OFFENCES AND PUNISHMENTS

The Examination Rules mentioned below refer to Examination Rules and Regulation Chapter XI of this manual.

8.1 EXAMINATION OFFENCES AND PUNISHMENT

1	Any candidate who violates Examination Rule 5 shall be deemed guilty of the offence of possession of unauthorized documents and shall be liable to cancellation of his candidature from the examination and to any further punishment that the Senate may decide upon.	Possession of unauthorized documents.
2	Any candidate who violates Examination Rule 7 shall be deemed guilty of the offence of copying and shall therefore be liable to cancellation of his candidature from the examination and to be prohibited from sitting any examination of the university for a period of time and to any other punishment that the Senate may decide	Copying

3	Any candidate who violates Examination Rule 8 shall be deemed guilty of the offence of having cheated at the examination and shall be liable to the cancellation of his candidature from the examination and to be prohibited from sitting any examination of the university for a period of not less than three years and to any further punishment that the Senate may decide.	Cheating or Plagiarism
4	Any candidate who is detected removing examination stationary and other material provided for the examination (Rule 10) shall be deemed guilty of an examination offence and shall be liable for punishment including cancellation and/ or prohibition from sitting any examination of the university for such period as may be specified by the Senate.	Removal of stationary
5	Any candidate who violates any one or more of the Examination rules 6, 14, 15, 16, 17 or 18 shall be deemed guilty of the offence of disorderly conduct and shall be liable to punishment including cancellation/ or prohibition from any examination of the university for such period as may be specified by the Senate	Disorderly conduct
6	Any candidate who violates Examination Rule 19 shall be guilty of the offence of impersonation and shall be liable to cancellation of candidature from the examination and to be prohibited from sitting any examination of the university for a period of not less than 5 years and to any further punishment that the Senate may decide. He/she may also be liable to any punishment under the penal code/ criminal law.	Impersonation
7	Any candidate who violates Examination Rule 20 shall be guilty of an examination offence and shall be liable to cancellation of candidature from the examination and to any further punishment that the Senate may decide upon.	Improper knowledge
8	Any candidate found aiding and abetting in the commission of any of the above examination offences shall be deemed to have committed that offence and shall be liable to the same punishments.	Aiding and Abetting

8.2 PROCEDURE DEALING WITH EXAMINATION OFFENCES BY CANDIDATES

1	There shall be an Examination Disciplinary Committee of not less than 3 members appointed annually, at the beginning of each Academic Year, by the Senate to enquire into and make recommendations (including punishments) into examination offences referred to it. Members should be from different faculties, to ensure that at least two members are from another Faculty when an inquiry is under process.	Examination Disciplinary Committee

8.3 PROCEDURE FOR REPORTING OF EXAMINATION OFFENCES AND PUNISHMENT

1	In all cases of violation of examination rules (Chapter XI) detected, the Supervisor shall take actions as outlined in this section and forward his report to the DR/SAR/AR of the faculty	Procedure for punishment of offences detected by the supervisor.

	The Supervisor’s report should be countersigned by one of the invigilators.	
2	In cases of disorderly conduct the supervisor shall in the first instance warn the candidate to be of good behavior. Disorderly conduct shall be considered grave, only if such conduct in the opinion of the supervisor is considered as causing a disturbance in the conduct of the Examination. Where the candidate persists in unruly or disorderly conduct the supervisor may exclude the candidate from the examination hall and issue him/her a letter with copies to the relevant Dean and DR/SAR Academic Affairs, cancelling his/her candidature from the examination. Where a candidate’s offence is only disobedience the supervisor shall warm the candidate and forward a report to Dean and DR/SAR Academic Affairs.	Cancellation of candidature for disorderly conduct
3	In all other cases of examination offences detected, the Supervisor shall on the detection of the offence take possession of unauthorized documents if any and obtain a statement from the candidate and write his report on the matter to the Dean of the faculty. Materials taken into custody shall be authenticated by placing the signature of the candidate and the Supervisor/invigilator and the date time and place of detection	Action to be taken by Supervisor.
4	The Dean after a preliminary inquiry shall place all reports of examination offences submitted by the Supervisors to the Exam Disciplinary Committee for further action	Refer to Exam Disciplinary Committee
5	Any examiner, Head of Department or any other official of the University who detects an examination offence, shall report the matter in writing to the Dean, who shall call for a preliminary inquiry and place the complaint to the Exam Disciplinary Committee for further action.	Offences reported by others.

8.4 FINAL DECISION ON EXAMINATION OFFENCES

1	The punishments recommended by the Examination Disciplinary Committee shall be submitted to the relevant Faculty Board for a decision and be referred to the Senate for ratification.	Decided by the Faculty Board and ratified by the Senate

8.5 APPEAL AGAINST PUNISHMENTS

1	There shall be an Appeals Board, consisting of three members, appointed by the Vice-Chancellor to consider the decisions made under Section 14.5.	Appeals Board appointed by the Vice-chancellor
2	Any student wishing to appeal against the punishment imposed on them should write to the Vice-chancellor in this regard within two weeks from the date of communication to them. The vice-chancellor shall consider the appeal and may decide to refer to the Appeals Board. Appeals Board shall either affirm or review the imposed punishment and make recommendation to the Vice Chancellor.	Appeal within two weeks to Vice-chancellor

9. DETAILED SYLLABUS OF BACHELOR OF SCIENCE IN COMPUTER SCIENCE

Year I Semester I

Course Code	CO1121	**Course Name**	Basic Mathematics for Computing		
Year	I	**Hourly Breakdown**	**Theory**	**Practical**	**Independent Learning**
Semester	I		30	-	70
Core/Optional	Core	**GPA/NGPA**	GPA		
Aim(s) / Objective(s):	This course is designed to provide students the basic mathematical and logical concepts to boost their mathematical thinking in a computing environment.				
Intended Learning Outcomes (ILOs):	At the end of the course, students will be able to: - define basics of mathematical concepts such as indices, logarithms and sets - define logical propositions, predicates and quantifiers - identify the solution using proof by direct, contradiction, counter examples, and mathematical induction - recognize techniques of counting				
Course Content:	**Indices and Logarithms:** Index laws, surds, e^x, Logarithms: Definition, laws of logarithms, change of base, Graphs of a^x, $\log_a x$; **Sets:** Introduction to sets , subsets, proper subsets, power sets, universal set, null set, equality of two sets, Venn diagrams, Set operations, Laws of algebra of sets proofs of the laws using Venn diagram, proofs of results using the laws.; **Logic :** Propositions, Propositional Logic, Arguments, Predicates and Quantifiers, Types of Proofs; **Relations:** Ordered pairs and the Cartesian product of two sets, Definition of a relation, Relation from a set A to a set B, relation on a set A, Relations as sets of ordered pairs Inverse of a relation, Directed graph, Equivalence Relations; **Functions:** Function as a mapping from a set A to a set B, Range of function: Function from a finite set A onto a set B, One to one functions, Bijections, Functions from a finite set A to a finite set B, Inverse functions, Composite functions, Graph of a function as a set of ordered pairs, Special functions and sketching their graphs; **Techniques of counting:** Permutations, Binomial theorem and the binomial coefficients, Combinations, Tree diagrams				
Teaching / Learning Methods:	Lecture-demonstration, Use of slides, take home exercises, tutorials, in-class activities				
Assessment Methods:	Formative Assessment - Problem sheets, Multiple choice questions, Structured questions Summative Assessment – Written Test				
Assessment Strategy:	Continuous Assessments - 35 % End-Semester Examination - 65 % Final Marks = Continuous Assessment + End Semester Examination				

Recommended Reading(s):	1. Discrete Mathematics: By A.Chtewynd and P.Diggle 2. Discrete Mathematics: By Olympia Nicodemi, CBS Publishers and distributors 3. Theory and problems of probability: By S.Lipshutz, McGraw Hill, Singapore 4. Theory and problems of finite Mathematics: By S. Lipshutz, McGraw Hill, Singapore 5. Elementary algebra for school: By Hall and Knight 6. Pure Mathematics: By Bostock and Chandler 7. Pure Mathematics: By Backhouse and Honldsworth Longman

Course Code	CO1122	**Course Name**	Basic Computer Programming		
Year	I	**Hourly Breakdown**	**Theory**	**Practical**	**Independent Learning**
Semester	I		30	-	70
Core/Optional	Core	**GPA/NGPA**	GPA		
Aim(s) / Objective(s):	This course is designed to introduce basic programming concepts and the object-oriented concepts				
Intended Learning Outcomes (ILOs):	At the end of the course, students will be able to: - recognize different programming languages and object oriented programming. - define variables, and datatypes. - identify various types of operators. - choose appropriate selective statements or iterative statements based on the problem.				
Course Content:	**Introduction to programming language:** Programming languages: Generations of languages, Translators, Program style and documentation, Basics of Object-Oriented Programming and its terminology Evolution, Introduction to Standard C++ Programming Language, Running C++Programs; **Statements Expressions, Variable and Data types:** Statements and Expressions, Comments in Standard C++, Literals, Variables, Data types; **Operators:** Arithmetic Operators; Logical Operators; Bitwise Operators; Relational Operators; Operator Precedence; **Arrays and Control Statements:** One Dimensional and Multidimensional Arrays, Array type for String, Selective Statements, Iterative Statements, Jump Statements; **Objects and Classes:** Definition of a class, Creating and destroying Objects, Defining methods, Parameter Passing: Passing arguments to methods, Constructor Methods: Overloading Constructors, Basic Operator Overloading, Friend Functions and Automatic type conversion, The Standard class String; **Pointers:** Introduction to Pointers, Dynamic Arrays, Classes, Pointers and Dynamic Arrays; **Separate Compilation, Namespaces; Input / Output Streams, Tools for I/O Stream; Inheritance; Recursive Functions; Templates; Linked Lists; Exception Handling; Iterators and Container; Generic Algorithms**				
Teaching / Learning Methods:	Lecture-demonstration, Use of slides, take home exercises, tutorials, in-class activities				
Assessment Methods:	Problem sheets, Multiple choice questions, Structures questions				
Assessment Strategy:	Continuous Assessments - 35 % End-Semester Examination - 65 %				

	Final Marks = Continuous Assessment + End Semester Examination
Recommended Reading(s):	1. Computer Programming: Fundamentals for Absolute Beginners by Alexander Bill, ISBN: 9781075569982 2. Fundamentals of C++ Programming by Richard L. Halterman

Course Code	CO1112	**Course Name**	Practical work on CO1122		
Year	I	**Hourly Breakdown**	**Theory**	**Practical**	**Independent Learning**
Semester	I		-	30	20
Core/Optional	Core	**GPA/NGPA**	GPA		
Aim(s) / Objective(s):	This course is designed to teach the students practical implementation of basic computer programming.				
Intended Learning Outcomes (ILOs):	At the end of the course, students will be able to: - demonstrate fundamental programming concepts - identify classes, objects, members of a class and relationships among them needed for a specific problem - develop codes to solve real world problems - provide solutions to mathematical problems using structural language - demonstrate the programming skills on problem-solving				
Course Content:	The practical implementation is based on the theory components covered in the course CO1122: Basic Computer Programming and the lab sessions will be based on the contemporary computer platforms and tools.				
Teaching / Learning Methods:	Handouts / Presentations, Laboratory experiments, activities,exercises, Practical records, Tutorial discussion				
Assessment Methods:	Group /Individual Presentations, Small Projects, Quizzes, Practical assessment tests to solve real world problems				
Assessment Strategy:	Continuous Assessments - 35 % End-Semester Examination - 65 % Final Marks = Continuous Assessment + End-Semester Examination				
Recommended Reading(s):	1. Computer Programming: Fundamentals for Absolute Beginners by Alexander Bill, ISBN: 9781075569982 2. Fundamentals of C++ Programming by Richard L. Halterman				

Course Code	CO1123	**Course Name**	Formal Methods for Problem Solving		
Year	I	**Hourly Breakdown**	**Theory**	**Practical**	**Independent Learning**
Semester	I		30	-	70
Core/Optional	Core	**GPA/NGPA**	GPA		
Aim(s) / Objective(s):	This course is designed to introduce systematic software development concepts using VDM				
Intended Learning Outcomes (ILOs):	At the end of the course, students will be able to: - define implicit specifications of functions, operations, and set notations, propositional logics, predicate logics - state whether statements are logically equivalent - state the truth value for predicates and quantifier statements - define concept of proof of basic mathematical problems - identify appropriate proof techniques to be used for a particular problem.				

Course Content:	**Logic of Propositions:** Propositional operators, Concept of Proof, Proofs in propositional calculus; **Reasoning about Predicates:** Truth valued functions, Quantifiers, Proofs in propositional calculus; **Functions and Operations:** Implicit specification of functions, Correctness proofs, Reasoning about partial functions, Implicit specification of operations; **Set Notation:** Set notation, Reasoning about sets, Theories of Datatypes, Specifications; **Co mposite Objects and Invariants:** Notation, Structural induction and invariants, States and proof obligations; **Map Notation:** Notation, Reasoning about Maps, Specification; **Sequence Notation:** Notation, Reasoning about Sequence, Specifications; **Data Reification and Data types:** Retrieve functions and adequacy, Operation modelling proof, Modules as data types, Exceptions, Implementation bias in models, Property oriented specifications of data types; **Operation Decomposition:** Decomposition rules, Assertions as annotations, Decomposition in Design, An alternative loop rules; **A Small Case Study:** Partitions of a fixed set, Specifications, A theory of forests, The Fischer/ Galler algorithm, Operation Decomposition
Teaching / Learning Methods:	Lecture-demonstration, Use of slides, take home exercises, tutorials, in-class activities
Assessment Methods:	Problem sheets, Multiple choice questions, Structures questions
Assessment Strategy:	Continuous Assessments - 35 % End-Semester Examination - 65 % Final Marks = Continuous Assessment + End Semester Examination
Recommended Reading(s):	1. Systematic Software Development Using VDM: By Cliff B. Jones, Second Edition, Prentice Hall Publications

Course Code	CO1124	**Course Name**	Computer Systems & PC Applications		
Year	I	**Hourly Breakdown**	**Theory**	**Practical**	**Independent Learning**
Semester	I		30	-	70
Core/Optional	Core	**GPA/NGPA**	GPA		
Aim(s) / Objective(s):	This course is designed to introduce basics of computer and communication technologies and their usages.				
Intended Learning Outcomes (ILOs):	At the end of the course, students will be able to: - identify various components of computer system and its functions - recognize different number systems and ways of data representation - relate the generations of computer - write basic DOS commands - select suitable application softwares (word, spreadsheet, database, and powerpoint) appropriate for different purposes - recognize computer networks				
Course Content:	**Introduction to Computers:** Functions of Computers (What is computer? Why computers needed?, What does a computer can do? Input / Output operations, Arithmetic and logical operations, Storage operations; Data / Information Processing., Components of Computer: Input / Output Units, Processing Units, Memory, Secondary storage devices; Functions of central processing unit, ALU, CU., Hardware / Software / Firmware., General and Special purpose Computers.), Computer System Overview: (Numbering Systems: Decimal, Binary, Base 4, Octal, Hexadecimal, data conversion, Data Representation: Character (Bit, Byte, KB, MB, GB, TB, ASCII, EBCDIC, Code), Number: Fixed Point, Floating Point, Data Transmission Word Length(8,16,32,64bits), Serial, Parallel, Logic Operation: NOT, AND, OR, NAND, NOR, XOR (using truth tables and gates)), Input and Out Put Devices (Keyboard, pointing devices (Mouse, track ball, touch pad, joystick), writing and Drawing Input Devices (Light pen, touch screen, digitizer) Video Input (Digital Camera), Text Input (Scanner, OCR), Voice Input (Voice Recognition), Text / Graphic / Sound / Video Output, Printer (Impact: - Daisy wheel, Dot Matrix Non-impact: - Laser, Inkjet), plotter, Monitor (Pixel, High and Low Resolution, Bit map, LCD),				

	Terminal (Dump, smart / intelligent)), Storage Devices **(**principles of Magnetic Drum and Tape, Floppy Disk, Hard Disk**,** Discussion on Seek time, Rotational Delay, Access time, Block Size, blocking factors, Inter blocking gap**,** RAID Devices, ZIP Drives, Digital Tape, CD ROM, DVD**),** Main Circuit Board of a PC **(**Chips, Ports, Expansion Slots, RAM, ROM, PROM, EPROM, EEPROM), Memory Hierarchy (Register, Buffer, RAM, Disk Cache, Disk, Tape (Capacity, Access time), Types of Processing: Batch, Real-Time, Online, offline, Computer Viruses and its Precautions); **History of Computers:** Evolution, 1to 5Generations, Classification of Computers (Old and Modern), PC Micro Processors: Intel Series 8 bit to Pentium; **MS DOS Operating Systems:** Single and Multiuser O/S, DOS Commands, Batch files: autoexec.bat, TSR routines, GUI; **Application Software:** Word-processing, Spreadsheet Applications, Database Applications; **Creation and Presentation of Computer Graphics:** Power Point; **Multimedia Tools and Devices**; **Introduction to PC Networks and Internets:** Evolution of Networks, Advantages of Networks, Components of Networks, The Internet, Intranet and Extranet
Teaching / Learning Methods:	Lecture-demonstration, Use of slides, take home exercises, tutorials, in-class activities
Assessment Methods:	Problem sheets, Multiple choice questions, Structures questions
Assessment Strategy:	Continuous Assessments - 35 % End-Semester Examination - 65 % Final Marks = Continuous Assessment + End Semester Examination
Recommended Reading(s):	1. Teach yourself all about computers: By Barry Press and Marcia Press 2. Using Computers and Information: By Jack B.Rochester

Course Code	CO1114	**Course Name**	Practical work on CO1124		
Year	I	**Hourly Breakdown**	**Theory**	**Practical**	**Independent Learning**
Semester	I		-	30	20
Core/Optional	Core	**GPA/NGPA**	GPA		
Aim(s) / Objective(s):	This course is designed to teach the students practicals on application softwares.				
Intended Learning Outcomes (ILOs):	At the end of the course, students will be able to: - do windows operating system - perform word processing - prepare spreadsheet - prepare presentations				
Course Content:	The practical implementation is based on the theory components covered in the course CO1124 Computer Systems & PC Applications and the lab sessions will be based on the contemporary computer platforms and tools.				
Teaching / Learning Methods:	Handouts / Presentations, Laboratory experiments, activities,exercises, Practical records, Tutorial discussion				
Assessment Methods:	Group /Individual Presentations, Small Projects, Quizzes, Practical assessment tests to solve real world problems				
Assessment Strategy:	Continuous Assessments - 35 % End-Semester Examination - 65 % Final Marks = Continuous Assessment + End Semester Examination				
Recommended Reading(s):	1. Teach yourself all about computers: By Barry Press and Marcia Press 2. Using Computers and Information: By Jack B.Rochester				

Course Code	CO1125	**Course Name**	Statistics for Science & Technology		
Year	I	**Hourly Breakdown**	**Theory**	**Practical**	**Independent Learning**
Semester	I		30	-	70
Core/Optional	Core	**GPA/NGPA**	GPA		
Aim(s) / Objective(s):	This course is designed to teach the students the statistical theories and its applications in the field of science and technology.				
Intended Learning Outcomes (ILOs):	At the end of the course, students will be able to: - identify samples and populations - recognize samples in terms of frequency distributions and central measurements - state various measure of dispersions, moments, skewness Kurtosis and apply probability theories - identify various standard distributions and apply elementary sampling theories - select different estimation parameters and apply statistical decision theories				
Course Content:	**Introduction:** Role of Statistics in Science and Technology, Types of data, Sample and sample statistics, Population and population parameters, Statistical inferences; **Frequency Distribution:** Class intervals, class limits, class boundaries, Histogram, Frequency polygon, Relative frequency distribution, Cumulative frequency distribution and Ogives, Frequency curves & smoothed Ogives, Types of frequency curves; **Measure of Central Tendency:** Statistical notations, Mean (Arithmetic mean, Weighed mean, Geometric mean, Harmonic mean), Median, Mode, Empirical relationship among mean, median and mode, RMS value, Quartiles, Deciles, percentiles; **Measure of Dispersions:** Dispersion or variation, Range, Mean deviation, semi-inter-quartile range, 10 to 90 percentile range, The standard deviation, Variance, Short method to calculate standard deviation, properties of standard deviation, Empirical relations between measures of dispersions, coefficients of variations, standard variation, standard scores; **Moments, Skewness and Kurtosis:** Moments and its relationships, moment computations, moments in dimensionless form, Population moments, Skewness, Kurtosis; **Probability Theory:** Definition, conditional probability, Independent and dependent Events, Mutually Exclusive events, Probability Distribution, Mathematical Expectation, Relation between population, sample mean and variance; **The Binomial, Normal and Poisson Distribution:** Binomial distribution, Normal distribution, Relation between the Binomial and Normal distribution, Poison distribution and the relation between Binomial and Poisson distribution; **Elementary Sampling Theory:** Random samples and sampling theory, Sampling with and without replacement, Sampling distribution of means, proportions, difference and sums, Standard Errors; **Statistical Estimation Theory:** Estimation of parameters, unbiased estimation, efficient estimation, Point estimation, Interval Estimation and their reliability, Confidence interval estimates of population parameters, Probable errors; **Statistical Decision Theory:** Statistical decisions, Hypotheses, Tests of Hypotheses and significance or decision rules, Type I and Type II Errors, Level of Significance, Two tailed, one tailed tests; **Small samples:** Student's t distribution, chi- square distribution, F-distribution, Chi-Square Test (Observed and theoretical frequencies, Definition of Chi square, Significance tests, the chi-square tests for Goodness of Fit, Contingency table, formula for computing chi-square); **Curve fitting and the method of least squares:** Relationships between variables, Equation of appropriate curves, Method of least squares, Non-linear relationships, The least squares of parabola, Regression; **Correlation Theory:** Correlation and regression, linear correlation, Measure of correlation, Multiple and partial correlation; **ANOVA:** Purpose, one way clarification, Short cut method for obtaining variance, Mathematical model for ANOVA, F-tests for the Null Hypothesis of equal measures				
Teaching / Learning Methods:	Lecture-demonstration, Use of slides, take home exercises, tutorials, in-class activities				

Assessment Methods:	Problem sheets, Multiple choice questions, Structures questions
Assessment Strategy:	Continuous Assessments - 35 % End-Semester Examination - 65 % Final Marks = Continuous Assessment + End Semester Examination
Recommended Reading(s):	1. Statistics Concepts and Applications by: Harry Frank & Steven C. Althoen 2. Mathematical Statistics by: J.N.Kapur, H.C.Saxena 3. Applied Statistics and Probability for Engineers by: Dougles,C.Montgomery, George C. Runger

Course Code	CO1115	**Course Name**	Practical work on CO1125		
Year	I	**Hourly Breakdown**	**Theory**	**Practical**	**Independent Learning**
Semester	I		-	30	20
Core/Optional	Core	**GPA/NGPA**	GPA		
Aim(s) / Objective(s):	This course is designed to teach the students practical implementation of statistical theories.				
Intended Learning Outcomes (ILOs):	At the end of the course, students will be able to: - demonstrate charts for samples in terms of frequency distributions and central measurements - apply different formulas for various measure of dispersions, moments, skewness Kurtosis and apply probability theories - identify various standard distributions and apply elementary sampling theories				
Course Content:	The practical implementation is based on the theory components covered in the course CO1125 Statistic for Science & Technology and the lab sessions will be based on the contemporary computer platforms and tools.				
Teaching / Learning Methods:	Handouts / Presentations , Laboratory experiments, activities, exercises, Practical records, Tutorial discussion				
Assessment Methods:	Group /Individual Presentations, Small Projects, Quizzes, Practical assessment tests to solve real world problems				
Assessment Strategy:	Continuous Assessments - 35 % End-Semester Examination - 65 % Final Marks = Continuous Assessment + End-Semester Examination				
Recommended Reading(s):	1. Statistics Concepts and Applications by: Harry Frank & Steven C. Althoen				

Course Code	CO1126	**Course Name**	Management Information System		
Year	I	**Hourly Breakdown**	**Theory**	**Practical**	**Independent Learning**
Semester	I		30	-	70
Core/Optional	Core	**GPA/NGPA**	GPA		
Aim(s) / Objective(s):	This course is designed to introduce different Management Information systems in the Information age and their applications.				

Intended Learning Outcomes (ILOs):	At the end of the course, students will be able to: - identify different MIS - recognize data warehouse and data mining - identify different database technologies - recognize various decision support and artificial intelligent systems - identify digital firms and system development cycles - state impact of IT on organisations, individuals, and society - recognize emergency trends and technologies
Course Content:	**Introduction:** What is MIS?, Importance and Evolution of MIS, Computers and MIS, Organisational Structures, Logical foundations of MIS, Types of Mis, Future of MIS; **The Information age and the changing the face of Business:** Today's Economy, New Economy, Global Economy, and Digital Economy, Information as a key resource, People as a key resource, Information Technology as a key resource, Roles and Goals of IT, Computer Hardware and Software(Categories of Computers by size: PDA, Notebook, Desktop, mini, mainframe, Supercomputer, Software: Application Software, System Software, Hardware: 1/O Devices, characteristics of CPU and RAM, Storage devices, Telecommunication devices and connectivity devices); **Using IT for competitive advantage:** Federal Express, Charles Sch web, Dell Corporation, Cisco Systems, Developing a strategy for the Internet Age: The five-force model and its usage, the three generic strategies and its usages, Bridging the gap between Business people and technical people, viewing business problem from another perspective, demanding a creative Design, the values chain, Looking beyond the company, Key E-Commerce Strategies, The U. S. Airline Industry (Airline Reservation System, Frequent Flyer Programs, Yield Management System), www, Search Engines (direct and true search engines), Ordering the sales product on the Internet, Websites, address, pages and understanding addressing and Brower software, Internet technologies: Backbone, Server's communications, Protocols; **Database Technology:** Database and Enterprise management, File processing System, Data independence, database approach, database architecture, DBMS, Data models, RDBMS, SQL, 4GL; **Databases and data warehouses:** Knowledge Management, The Relational Database model, DBMS tools, Data warehouses and Data mining : Analysing and Visualization, Managing the information resources in an organisations, Building Information Systems: Designing and Building a relational Database; **Decision support and Artificial Intelligence:** Decision support systems, Collaboration systems, Geographic Information System, Artificial Intelligence, Expert systems, Neural Networks, Genetic Algorithms, Intelligent Agents; **The Digital Firm: Electronic Business and Electronic Commerce:** Doing Business in the Digital Economy E-Business, Use of EDI and Extranets in E-Business, Growth of E-Commerce, Advantages, Keys to Success in Business to Customer E-Commerce, Business to Business E -Commerce, E- Commerce Payment Systems, Role of E-Government, Key E- Commerce Strategies, The U.S Airline Industry is an example; **System Development: Steps, tools and techniques:** Seven phases in the Systems Development Life Cycle, Knowledge workers and their roles in SDLC, Why system fails?, Self-sourcing and Outsourcing, Proto typing; **IT infrastructure: Business -Driven Technology:** Organisational goals and strategies (Increase employee's productivity, Enhance decision making, Improve team collaboration, Create business partners and alliances, Enable global reach, Facilitate original transformation), IT infrastructure and the real world, Impacts of IT on Organizations, Individuals, and Society; **Protecting people and information:** Threats and Safeguards: Ethics, Privacy, Information, Security and controls, Computer Crimes and Forensics: Computer Crimes, Computer Forensics, Recovery and Interpretation, Social issues in the Digital Firms; **Emergency Trends and Technologies:** Business, People, and Technology tomorrow, The need for information filtering, The movement towards intellectual, Computing, Changing in Physiological interaction, Increasing portability and mobility, The Digital Frontiers, The rebirth of E-Commerce and other important considerations

Teaching / Learning Methods:	Lecture-demonstration, Use of slides, take home exercises, tutorials, in-class activities
Assessment Methods:	Problem sheets, Multiple choice questions, Structures questions
Assessment Strategy:	Continuous Assessments - 35 % End-Semester Examination - 65 % Final Marks = Continuous Assessment + End Semester Examination
Recommended Reading(s):	1. Management Information System for the Information Age By: Stephen Haag, Maeve Cummings and Donald J. McCabbrey 4th Edition, Tata McGraw Hill 2. Information Technology for Management Transforming Organizations in the Digital Economy By: EFRAIM TURBAN, EPHRAIM MCLEAN and JAMES WETHERBE 3. Management Information System By: S. Sadagopan, Prentice Hall of India, 2001 4. Management Information System Managing the Digital Firm By: Kennerth C. Laudon and Jane P, Laudon 5. Management Strategy for I.T. An international Perspective By: Wendy Curie, Pitman Publishing

Course Code	GEP - I	**Course Name**	General English Proficiency - I		
Year	I	**Hourly Breakdown.**	**Theory**	**Practical**	**Independent Learning**
Semester	I		30	-	70
Core/Optional	Core	**GPA/NGPA**	NGPA		
Aim(s) / Objective(s):	This course is designed to teach the students English skills for understanding the technical environment.				
Intended Learning Outcomes (ILOs):	At the end of the course, students will be able to: - comprehend what they listen to in English - use the spoken form in their day to day activities - produce good technical writing - use electronic media for learning English				
Course Content:	**UNIT 1: Listening:** Introducing learners to GIE - Types of listening - Listening to audio (verbal & sounds); **Speaking:** Speaking about one's place, important festivals etc. - Introducing oneself, one's family / friend; **Reading:** Skimming a reading passage - Scanning for specific information - Note-making; **Writing:** Free writing on any given topic (My favourite place / Hobbies / School life, etc.) - Sentence completion – Autobiographical writing (writing about one's leisure time activities, hometown, etc.); **Grammar:** Prepositions - Reference words - Wh-questions - Tenses (Simple); **Vocabulary:** Word formation - Word expansion (root words / etymology); **E-materials:** Interactive exercises for Grammar & Vocabulary - Reading comprehension exercises - Listening to audio files and answering questions; **UNIT II: Listening:** Listening and responding to video lectures / talks; **Speaking:** Describing a simple process (filling a form, etc.). Asking & answering questions. Telephone skills - Telephone etiquette: Reading - Critical reading Finding key information in a given text Sifting facts from opinions; **Writing:** Biographical writing (place, people) - Lab descriptions (general / specific description of laboratory experiments) - Definitions - Recommendations; **Grammar:** use of imperatives - Subject-verb agreement; **Vocabulary:** Compound words - Word Association; **E-materials:** Interactive exercises for Grammar and Vocabulary - Listening exercises with sample telephone conversations/lectures - Picture-based activities.				

Teaching / Learning Methods:	Direct Interaction , Online Resources, Self Study
Assessment Methods:	Group activity, Written Test
Assessment Strategy:	Continuous Assessments - 35 % End-Semester Examination - 65 % Final Marks = Continuous Assessment + End Semester Examination
Recommended Reading(s):	1. Mindscapes: English for Technologists and Engineers, Department Orient Black Swan, 2012 2. S.P. Dhanavel, English and Communication Skills for students of Science and Engineering, Oriented Black Swan, Chennai, 2011 3. Pickett, Nell Ann, Ann A. Laster and Katherine E. Staples : Technical English: Writing, Reading , and Speaking, New York: Longman, 2011 4. Savarimuthu, J. S. Rohan and G. Petricia Alphine Nirmala. *English Grammar and Usage: An Ideal Companion for Advanced Learners*. Chennai: New Century Book House (NCBH), June 2016. (ISBN 978-81-2343-204-5) (Code No. A3506)

Year I Semester II

Course Code	CO1221	**Course Name**	System Analysis and Design		
Year	I	**Hourly Breakdown**	**Theory**	**Practical**	**Independent Learning**
Semester	II		30	-	70
Core/Optional	Core	**GPA/NGPA**	GPA		
Aim(s) / Objective(s):	This course is designed to introduce System analysing and development of designing a suitable model for a practical problem.				
Intended Learning Outcomes (ILOs):	At the end of the course, students will be able to: - recognize different system development approaches - compare different system development life cycles - identify system requirements - design and illustrate a suitable system - understand object-oriented development methods - discuss CASE Tools and their benefits				
Course Content:	**Introduction to System Design Environment:** System Development Approaches (Function Oriented, Data Oriented, Object Oriented), Development Process (Methodologies, Tools, Modelling methods), Processing Types and systems (Batch Processing, Real Time Processing), Management Process (Management, System Analysts, Programmers, Computer Operators, End Users), System Structure (People, Processes, and Data, Databases), Personal Systems, Centralized Systems (Data warehousing, data mining), Distributed Systems (Evolution of Distributed Processing, Client Server Systems, Agent Oriented Systems); **System Development Life Cycle (SDLC):** Linear and water fall cycles (Linear Cycle phases: Problem definition, system specification, system design, system development, testing, maintenance), Problems with Linear Cycles, Interactive cycles (Spiral Model); **Requirements Analysis:** Importance of communication, Identifying requirements (Data and Facts gathering techniques, Feasibility studies), Introduction to prototyping (Rapid prototyping tools, Benefits of prototyping); **System Design and Modeling:** Logical and Physical Design, User interface Design (Interface Design tools, User interface evaluations), Introduction to Process modelling, Introduction Data				

	Modeling; **System Design Techniques:** Document Flow Diagrams (Documents, Physical Movement of Documents, Usefulness of Document Flow Diagrams), Data Flow Diagrams (DFD Notation, Context Diagram, DFD levelling (Process Descriptions, Structured English, Decision Trees, and Decision Tables)), Entity Relationship Diagrams (Entities, Attributes, Relationships, Degree, Optionality, Resolving many to many relationships, Exclusive relationships), Structure charts (Modules, Parameter Passing, Execution sequence, Structured Design, Conversion from Data Flow Diagrams to Structure Charts); **Introduction to Object Modeling and Object Development methods:** Representations: Classes, Objects, Associations, Aggregations, Inheritance, Multiple Inheritance, Modelling behaviour: Actors, Use cases, Interaction diagrams, State diagrams, Object Development Methods: (Methodologies: OOSE (Object Oriented Software Engineering); Grady Booch A Design Method, OMT (Object Modeling Techniques) Method; Rational Unified Process (RUP), Object Libraries: Reuse, Continual Refinement); **System Implementation, Maintenance and Documentation:** Testing, Evaluation, Maintenance Activities, Documentation (Document Configuration, Maintaining a configuration); **CASE Tools:** Computer Aided Software Engineering: Methodologies, Techniques and Tools, Components of a CASE tool: (Diagramming Tools, Report Generators, Information Repository, code Generators, Benefits of Using CASE Tools in System Development)
Teaching / Learning Methods:	Lectures, Tutorials, class discussions, take home exercises, Guided learning
Assessment Methods:	MCQ, Structured Question, Presentation
Assessment Strategy:	Continuous Assessments - 35 % End-Semester Examination - 65 % Final Marks = Continuous Assessment + End-Semester Examination
Recommended Reading(s):	1. System Analysis and Design Methods by: Jeffrey L. Whitten, Lonnie D. Bentey (Tata McGraw-Hill) 2. Practical SSADM: A complete Tutorial Guide, Philip L Weaver (Pitman Publishing) 3. An Introduction to System Analysis Techniques, Mark Lejk, David Deeks (Prentice Hall) 4. System Analysis and Design, Don Yeates, Maura Shields and David Helmy (Longman group Ltd)

Course Code	CO1222	**Course Name**	Data Structures & Algorithms		
Year	I	**Hourly Breakdown.**	**Theory**	**Practical**	**Independent Learning**
Semester	II		30	-	70
Core/Optional	Core	**GPA/NGPA**	GPA		
Aim(s) / Objective(s):	This course is designed to teach various data structures and standard algorithms that are used to solve various real-world problems.				
Intended Learning Outcomes (ILOs):	At the end of the course, students will be able to: - recognize various data structures and algorithms - relate them to solve real world problems - classify different algorithms - understanding sorting and searching algorithms - estimate time complexity for different algorithms				
Course Content:	**Introduction to Data Structures:** Definition of Data Structures, Static and Dynamic Implementation, Examples of real-life applications; **The Stacks:** Definitions, Array based and Linked List implementation, Examples: Infix, Postfix, Prefix representation, Application: A simple calculator, Mathematical Expression Evaluation; **Queues and Lists:** Definition, Array				

	based/ Linked List implementation, Circular implementation of Queues and Singly/ Doubly linked list, Applications; **Trees:** Definition of Trees and Binary Trees, Properties of Binary Trees and Implementation, Binary Traversal: Pre-Order, Post-Order, In-Order Traversals, Binary Search Trees Implementations, Balanced Trees, AVL Trees, Implementations; **Graphs:** Definition of Undirected and Directed Graphs, Array based implementation of graphs, Adjacency Matrix, Path Matrix implementation, Linked List representation of graphs, Shortest Path Algorithm, Graph traversal: Breadth first and Depth first traversals, Connectivity of Graphs, Applications; **Tables:** Definitions, Hash function, Implementation and Applications; **Running Time:** Time Complexity: Big O notation, Running times: best case, worst case and average case, Factors depends on running time, Introduction to Recursion, Divide and Contour Algorithm, Evaluating time Complexity; **Sorting Algorithms:** Basic sorting algorithms: Bubble sort, Selection Sort, Insertion Sort and their implementations, Efficiency of the above algorithms, Recursive Algorithms: Shell Sort, Merge Sort, Quick Sort algorithms, Heap Sort, Radix sort algorithms; **Searching Algorithms:** Straight Sequential Search: Array and Linked List Implementation, Binary Search: Recursive and Non-recursive algorithms, Indexed Sequential Search
Teaching / Learning Methods:	Lectures, Tutorials, class discussions, take home exercises, Guided learning
Assessment Methods:	MCQ, Structured Question, Presentation, Oral questions
Assessment Strategy:	Continuous Assessments - 35 % End-Semester Examination - 65 % Final Marks = Continuous Assessment + End Semester Examination
Recommended Reading(s):	1. Analysis of Algorithms: by M.A. Weiss 2. Data Structures and Algorithms: by A.V.Aho

Course Code	CO1212	**Course Name**	Practical work on CO1222			
Year	I	**Hourly Breakdown**	**Theory**	**Practical**	**Independent Learning**	
Semester	II		-	30	20	
Core/Optional	Core	**GPA/NGPA**	GPA			
Aim(s) / Objective(s):	This course is designed to teach the students practical implementation of Data Structures & Algorithms theories.					
Intended Learning Outcomes (ILOs):	At the end of the course, students will be able to: - Implement various data structures and algorithms - Relate them to solve real world problems					
Course Content:	The practical implementation is based on the theory components covered in the course CO1222 Data Structures & Algorithms and the lab sessions will be based on the contemporary computer platforms and tools.					
Teaching / Learning Methods:	Demonstrations, Problem Sheets					
Assessment Methods:	Group /Individual Presentations, Small Projects, Quizzes, Practical assessment tests to solve real world problems					
Assessment Strategy:	Continuous Assessments - 35 % End-Semester Examination - 65 % Final Marks = Continuous Assessment + End Semester Examination					
Recommended Reading(s):	1. Analysis of Algorithms: by M.A. Weiss 2. Data Structures and Algorithms: by A.V.Aho					

Course Code	CO1223	**Course Name**	Data Base Management Systems		
Year	I	**Hourly Breakdown**	**Theory**	**Practical**	**Independent Learning**
Semester	II		30	-	70
Core/Optional	Core	**GPA/NGPA**	GPA		
Aim(s) / Objective(s):	This course is designed to introduce the fundamentals of DBMS and their practical usages.				
Intended Learning Outcomes (ILOs):	At the end of the course, students will be able to: - understand the file organisation and access mechanisms - interpret DBMS evolution, data models, design models - illustrate ER diagram - apply normalisation techniques - manipulate data				
Course Content:	**File organisation and access mechanisms (6hrs.):** Introduction Physical Storage of Data (Guidelines for physical database design, Concepts of indexes.; Secondary-Key Access: Primary key, Secondary key.; File organisation terms and concepts; Operations on Files; Keyed and non-keyed files.; Physical Access of the Database: Strategy selector, buffer manager, file manager, Disk storage.; Physical Storage media: Main Memory, Secondary storage, disk drive, cylinder, structure of a disk surface, Magnetic Disk system, Physical storage blocks, Block addressing.), Performance Factors, Storage Formats and File Organisation Methods (Disk Performance Factors: Access motion time, head activation time, rotational delay, data transfer rate, data transfer time, example of a random access and sequential access records., Data Storage Formats: Track formats, Count-key format, Count-data format, Record formats, Physical record, Fixed-length records, Variable-length records, Input/ Output Management, Introduction to file organisation methods: Heap, ISAM, Hashing, B-Tree), File Organization and Addressing Methods (4hrs.) (Serial File, Sequential File Organization, Index-Sequential File Organization; illustrations., Direct File organisation, Static Hash Functions, Dynamic hash function, illustrations, Implementing Logical Relationships: Linked lists, Pointers, Head list, Inverted lists, Balanced-Tree index (B+ Tree), Rooted tree, leaf, path, Construction of B-Tree index), Mapping Logical Data Structures to Physical Data Structures File organisation for relational tables. (Mapping process for Relational Data structures, clustering of Tables, Clustering indexes., De-normalisation.), Database Administration and Control (Data administrator, Database Administrator (DBA), Functions of a DBA., Roles of a DBA with respect to Database Integrity, Transaction Processing Concurrency Control, Database Security and Database Recovery.); **Introduction to DBMS:** The Evolution of Database Technology (Data, information, database, database system, database management system; increasing use of data as a corporate resource, Data processing and data management., File oriented systems: Meeting the need for random access processing; Limitations of Traditional File Systems: Data redundancy, Inadequate data manipulation capabilities, program data dependency; Data independence.), Database Architecture (Components of a Database Management System (DBMS): Data Dictionary (importance, contents), meta data; Data security and integrity mechanism; Concurrent access for multiple users; User oriented data query and reporting Application development facilities., Database Systems; ANSUSPARC Three-level Architecture: Conceptual model, Logical model, Physical model, External view, Conceptual view, Internal view of data., Data specification and access mechanisms: Data Definition Language (DDL), Sub-Schema DDL (SDDL), Data Manipulation Language (DML); Users: End users, Database Administrator (DBA); DBMS: Functions, Capabilities, Advantages and disadvantages.); **Data Model (02 hrs):** Introduction to Data models (Brief overview of Hierarchical, Network, Relational, Object relational and Object oriented data models Outline of the Data definition and data manipulation constructs in each of the above data models Comparison of the above data models), Introduction to Current Directions (Database Server, Client/Server Platforms, Distributed Databases Data Warehousing and Data Mining Open Systems, Interoperability, Database access over Internet, Open Database Connectivity				

	(ODBC)); **Database design process (05 hrs):** Database Design Approach (Introduction: Benefits, Critical success factors, Where it fits into the application development process, Approach, Data requirement analysis: Gain an understanding of the business; Conceptual modelling: Identify the principal data objects, Diagram the data objects using the entity-relationship (ER) approach, Resolve the conceptual data model, Determine attribute specifications and data types, Verify the conceptual data model through normalisation; Logical model; Physical model; Database Design tools.), ER Concepts and Terminology (Three classes of objects: Entities, Relationships and Attributes., Entities: Entity, Entity instance, Subtype and Supertype Entities, Strong and weak entities, Generalisation, specialisation and aggregation., Relationships: Connectivity (binary, n-ary), (1:1, 1:N, M:N), Determining the connectivity, Cardinality, Existence dependency (mandatory, optional)., Attributes: Identifying attributes, Attribute types (identifier, descriptor), Derived data, Domain, Composite attributes.), ER Diagrams (The Role of ER Diagrams., Basic Objects: Bachman Style, Relationship Representation., Alternative Syntax: Chen., Exercises.), Mapping Conceptual model into relational schema (2hrs.) (Regular, weak, generalised and specialised entities, Relationship types, Multi-valued attributes., Resolve the conceptual data model; Redundant Relationships; Recursive Relationships; Resolving Relationships: 1:1, M:N.), Attribute Specifications and Data types (3hrs.) (Attribute names, Naming conventions, Avoid Synonyms and Homonyms, Null Values, Entity integrity, Unique Requirement., Categories of Data Types: Character, Numeric, Variable Character, Date, Serial, Money, Datetime, Interval., Character: CHARACTER (CHAR); Numeric: INTEGER (INT), SMALLINT, FLOAT, SMALLFLOAT, DECIMAL; Variable Character: CHARACTER VARYING (VARCHAR); Binary Large Object (BLOB): Text, Byte.); **Data normalisation process and the normal forms (02hrs.):** Introduction to data normalisation and normal forms (What is normalisation, Benefits of normalisation, Normalisation Rules 1NF, 2NF, 3NF and Higher NF.), First Normal Form (1NF, Why convert to INF, Conversion to 1NF;), Second Normal Form (2NF, Functional Dependence and Fully Functional Dependence, Why convert to 2NF, Conversion to 2NF), Third Normal Form (3NF, Transitive Dependence, Why convert to 3NF, Conversion to 3NF.), Normalisation considerations (Good and bad decompositions Multi-valued dependencies, Join dependencies. Higher Normal Forms: Boyce-Codd NF, 4NF, SNF, Domain-Key NF); **Data Manipulation (12hrs.):** Relational Data Model (Introduction: Review of Logical data models, Definition of Relation, properties, tuple, domain, instance, cardinality, degree, schema. Concepts of keys: Candidate key, Primary key, Alternate key, Composite key, Surrogate key, Foreign key. Fundamental integrity rules: entity integrity, referential integrity.), Procedural Query Languages (Introduction: Different forms of query language interfaces; Query By-Example (QBE), Graphical, Procedural and Declarative query Languages., Mathematical foundations; Prepositional and predicate calculus: Boolean Algebra laws, Truth-valued function, free and bound variables, Precedence rules for the connectives, constant, variable, function reference., Relational algebra (RA): Traditional Set Operations (Union, Intersection, Difference, Product), Special Relational Operations (Select or Restrict, Project, Join, Different types of join (theta join, equi join, natural join, outer joins). Divide), Minimal set of operations, Simple and Complex queries using RA.), Declarative Query Languages (Relational calculus (tuple-oriented): target list, qualifying statement, Quantifiers (EXISTS, FOR ALL), relational algebra vs relational calculus. Structured Query Language (SQL); Introduction to SQL standards: SQL86, SQL89 and SQL92.), Creating SQL Databases and Tables (Creating a Database: CREATE DATABASE, Creating a database schema; Database options: Connect, Disconnect, Select, Close, Create, Drop., Defining tables and views: CREATE TABLE, ALTER TABLE, DROP TABLE, Specifying integrity constraints: PRIMARY KEY, UNIQUE, NOT NULL, CHECK, Referential Integrity constraints (Cyclic, Self referencing, Multiple path) FOREIGN KEY (CASCADE, RESTRICT, NULIFIES), DEFAULT.), Selecting Data (Queries: SELECT Statement. (Single Table: all columns (*), selecting specific columns (RA project operation), unique values (DISTINCT), Executing multiple statements (;),WHERE clause (RA select operation), including or excluding rows (=, !=), Relational Operators (=, !=, >, >=, <, =<), Identifying Null values (IS

	NULL), Where clause keywords (AND, OR, [NOT] BETWEEN, [NOT] IN, IS [NOT], NULL, [NOT] LIKE, ORDER BY, Arithmetic Operators (+, -, *, /), Expressions, Display Labels, Aggregate Functions: COUNT, SUM, AVG, MAX, MIN, GROUP BY, HAVING.), Multiple Table: RA join and product operations, Natural Join, Multiple Table Joins, Aliases for table names, Outer Join, UNION., Functions: Arithmetic (ROUND, TRUNC), String (TO CHAR, UPPER, LOWER, Sub strings, Concatenation, TRIM), Date and Time (DAY, MONTH, YEAR, DATE, CURRENT)., Sub queries: Nested Select Statement, Values returned by sub queries (single value, a list of values), EXISTS, Correlated nested queries); **Data Insertion, Updating and Deletion:** (Inserting Data: INSERT INTO. [VALUESISELECT] including a column list, null values; obtaining values from a SELECT., Updating Data: UPDATE (selected columns, selected rows, with a sub query)., Deleting Data: DELETE (all data, selected data, with a sub query)., Insert Data from ASCII operating system file and Write Data to ASCII operating system file); **Data View and Security:** (Characteristics of user views, View definition and use: .Database CREATE VIEW, DROP VIEW Security: GRANT, REVOKE); **Optimising Queries:** Guidelines to optimise queries, Creating indexes: CREATE INDEX, DROP INDEX, Temporary tables; Optimising queries with Selection, Projection and Join operations.; **Introduction to Database Transaction and Recovery:** Standard-alone and embedded query languages., Triggers and events; stored procedures., Transactions: Concepts of transactions and transaction processing, COMMIT and ROLLBACK., Database concurrency and database recovery: Ill effects of concurrency, transaction logs, concepts of two-phase locking, deadlocks.; **Introduction to 4GL Development Environment:** Overview of GUI design; Designing menus, screens and reports; data validation in data entry screens., Creating Databases and Tables; Creating and using Forms, Queries and Reports.
Teaching / Learning Methods:	Lectures, Tutorials, class discussions, take home exercises, Guided learning
Assessment Methods:	MCQ, Structured Question, Presentation, Oral questions
Assessment Strategy:	Continuous Assessments - 35 % End-Semester Examination - 65 % Final Marks= Continuous Assessment + End-Semester Examination
Recommended Reading(s):	1. Database Management and Design by G.W.Hansen and J.V.Hansen, Prentice-Hall 2. Database System Concepts by A.Silberschatz McGraw-Hill 3. Principles of Database Management by A.K. Majumdar and P. Bhattacharyya, McGraw-Hill 4. Fundamentals of Database Systems by R.FElmasri and S.B.Navathe

Course Code	CO1213	**Course Name**	Practical work on CO1223		
Year	I	**Hourly Breakdown**	**Theory**	**Practical**	**Independent Learning**
Semester	II		-	30	20
Core/Optional	Core	**GPA/NGPA**	GPA		
Aim(s) / Objective(s):	This course is designed to teach the students practical data organising, storing and manipulation using a Database management system.				
Intended Learning Outcomes (ILOs):	At the end of the course, students will be able to: - Create, modify and update tables. - perform queries for data retrieval - handle databases				
Course Content:	The practical implementation is based on the theory components covered in the course CO1223 Data Base Management Systems and the lab sessions will be based on the contemporary computer platforms and tools.				

Teaching / Learning Methods:	Demonstration, Problem sheets
Assessment Methods:	Group /Individual Presentations, Quizzes, Practical assessment tests to solve real world problems
Assessment Strategy:	Continuous Assessments - 35 % End-Semester Examination - 65 % Final Marks = Continuous Assessment + End-Semester Examination
Recommended Reading(s):	1. Fundamentals of Database Systems by R.F. Elmasri and S.B.Navathe 2. Database System Concepts by A.Silberschatz McGraw-Hill

Course Code	CO1224	**Course Name**	Multi Media & Hyper Media Development		
Year	I	**Hourly Breakdown.**	**Theory**	**Practical**	**Independent Learning**
Semester	II		30	-	70
Core/Optional	Core	**GPA/NGPA**	GPA		
Aim(s) / Objective(s):	This course is designed to introduce fundamentals of Multimedia and Hypermedia techniques and the practical implementations.				
Intended Learning Outcomes (ILOs):	At the end of the course, students will be able to: - Outline the hardware and software requirements for multimedia & hypermedia developments - recognize various file types and data compression techniques - describe multimedia applications in the Internet - discuss multimedia applications - identify social and legal issues in the multimedia applications				
Course Content:	**Introduction to Multimedia & Hypermedia (02hrs):** What is Multimedia and Hypermedia?, Interactivity and Non-linearity, Origin of Hypermedia and HTML; **Analog Vs Digital Systems (04hrs):** Continuous and Discrete signals, Sampling techniques, Data Volume and resolution, Data Transferring Techniques (DMA Vs PIOO, RAID and Bus Mastering technology, Fire Wire); **Hardware that Enables Multimedia (04 hrs):** CRTs and LCD Panels, MID Protocol, Sound Cards, Speaker Formations and Microphones, Video Capture Cards, Video Cameras, and Frame Grabbers, Joysticks, Drawing Tables and HMDs, Digital Cameras and Scanners, CD-ROM and DVD-ROM Technology, Home Consoles; **File Types, their Features and Usage (12hrs):** Text Formats, Graphics File Types, Audio File Types and Audio Compression (MP3/ ADPCM), Videos File Types (AVI, MOV, ASF) and Video Compression (MPEG, RLE, etc.); **Authoring Multimedia (08hrs):** Design Considerations (Human Computer Interaction Fundamentals, Foundations of Interactivity Design, Design rules for Graphic and Screen Design, Preventing and Handling Errors: System and Human), Audit Editing and MIDI Equipment, Video Editing, Hybrid CD, VCD, DVD Development, Object Oriented Development Environments, Multimedia Authoring Tools; **Virtual Reality (02hrs):** Hardware and Software Requirements, Levels of VVR, Applications if VR; **Multimedia in the Internet (12hrs):** Bandwidth and File Size Issues, Streaming Technology, SMIL and XHTML, Vector Animations, Java Scripts and Java Applets; **Emerging Trends and the Future (02hrs):** The digital Convergence, Interactive Television, Video-on-Demand Technology, Hybrids PDAs, Digital Paper, Home of the future; **Social and Legal Issues (02hrs):** Working Habits and their Social Significance (Tele working- Advantages and Implications), Social Interaction and Virtual Communities, Intellectual Property, Social Policies and Copyrights				
Teaching / Learning Methods:	Lectures, Tutorials, class discussions, take home exercises, Guided learning				
Assessment Methods:	MCQ, Structured Question, Presentation				

Assessment Strategy:	Continuous Assessments - 35 % End-Semester Examination - 65 % Final Marks = Continuous Assessment + End-Semester Examination
Recommended Reading(s):	1. Tannenbaum Robert S., Theoretical Foundations of Multimedia, W.H.Freeman and Company, 1998, ISBN 0-7167-8321-5x 2. Cawkell T., The Multimedia Book, Routledge, 1996, ISBN: 0-415-13666-0 3. R.S Tahnenbanm, Theoretical Foundation of Multimedia I. A/W that enables Multimedia II. Design considerations 4. John F. Hoegel Buford, Multimedia systems, I. Video Technology II. Digital Video and Image Compression 5. Garrand T, Writing for Multimedia, Butterworth-Heinemann, 1997, ISBN: 0-24080247-0 6. Keyes J. (Ed), The Ultimate Multimedia Handbook, McGraw-Hill, 1997, ISBN:0-07-034530-9 7. Vaughan T. Multimedia Making IT Work, McGraw-Hill, 1997, ISBN: 0-07-882225-4 8. Solari, Stephen J., Digital Video and Audio Compression, McGraw-Hill, 1997, ISBN 0-07-059538-0 9. Ralf Steinmetz and Klara Nahrstedt for Multimedia Computing, Communications and Applications, Pearson Education Asia, 2001.

Course Code	CO1214	**Course Name**	Practical work on CO1224		
Year	I	**Hourly Breakdown**	**Theory**	**Practical**	**Independent Learning**
Semester	II		-	30	20
Core/Optional	Core	**GPA/NGPA**	GPA		
Aim(s) / Objective(s):	This course is designed to teach the students practicals on Multimedia & HyperMedia Development tools.				
Intended Learning Outcomes (ILOs):	At the end of the course, students will be able to: - design contents using multimedia technologies - design multimedia contents in a webpage				
Course Content:	The practical implementation is based on the theory components covered in the course CO1224 Multimedia & HyperMedia Development and the lab sessions will be based on the contemporary computer platforms and tools.				
Teaching / Learning Methods:	Handouts / Presentations , Laboratory experiments, activities,exercises, Practical records, Tutorial discussion				
Assessment Methods:	Group /Individual Presentations, Small Projects, Quizzes, Practical assessment tests to solve real world problems				
Assessment Strategy:	Continuous Assessments - 35 % End-Semester Examination - 65 % Final Marks = Continuous Assessment + End-Semester Examination				
Recommended Reading(s):	1. Z.N. Li and M.S. Drew, “Fundamentals of Multimedia” 2. HTML and CSS: Design and Build Websites – by Jon Duckett 3. A Smarter Way to Learn JavaScript by Author: Mark Myers				

Course Code	CO1225	**Course Name**	Computer Architecture		
Year	I	**Hourly Breakdown**	**Theory**	**Practical**	**Independent Learning**
Semester	II		30	-	
Core/Optional	Core	**GPA/NGPA**	GPA		

Aim(s) / Objective(s):	This course is designed to introduce fundamental principles behind computer architecture.
Intended Learning Outcomes (ILOs):	At the end of the course, students will be able to: - compare and contrast fundamental elements and operations of computer - distinguish general system architecture, instruction set architecture and pipelined CPU architecture - describe memory hierarchy and I/O techniques - relate instruction parallelism and processor level parallelism
Course Content:	**Fundamental Principles (7hrs):** Radix number systems (Decimal, Binary, and Hexadecimal number systems, Binary arithmetic: addition, complements, and subtraction), Binary Codes (BCD code, ASCII character code), Boolean algebra and Logic Gates: (Boolean functions, Logic Gates: AND, OR, NOT, NOR, NAND, XOR, Simplification of Boolean functions: (2,3, and 4 variable Karnaugh maps)), Combinational logic (Adders, Multiplexors, Encoders), Sequential logic, Latches (Flip-Flops, Registers, Counters); **General System Architecture (4hrs.):** Flynn's classification (SISD, MISD, MIMD), Stored program control concept, Von Neumann architecture, Multilevel viewpoint: from Hardware to ISA level, Structural organization (an overview) (CPU, caches, main memory, secondary memory), Performance metrics (MIPS, MFLOPS, word length); **Instruction Set Architecture (5hrs.):** Instruction set based classification (RISC, CISSC, RISC vs CISC comparison), Addressing modes: (Instruction set could be generic RISC or x86) (Register, immediate, direct, indirect, indexed), Operations in the instruction set (Arithmetic and Logical (Add, Subtract, And, Or), Data Transfer (Loads and Stores), Control Flow (Branch, Jump, Procedure Call and Return, Traps)), Instruction set formats (Fixed, Variable, Hybrid); **Basic non-pipelined CPU Architecture (4hrs.):** CPU architecture types (Accumulator, register, stack, memory/register), Detailed data path of a typical register-based CPU, Fetch Decode-Execute cycle (typically 3 to 5 stage), Microinstruction sequencing (within each stage above), Implementation of control unit (microprogramming and hard-wired control options), Calculation of CPI and MIPS parameters; **Memory Hierarchy & 1/0 Techniques (6hrs.):** Memory hierarchy (Locality of reference principle, Memory hierarchy in practice: Cache, Main memory and Secondary, memory, Memory parameters: access/cycle time, cost per bit), Main memory (Semiconductor RAM & ROM organisation, memory expansion, Static & Dynamic memory), Cache memory (Associative & direct mapped organisations), Secondary Memory (Magnetic Disks, SCSI Disks, CD-ROMS), I/O methods (Programmed, Interrupt driven & direct memory access); **Introduction to Parallelism (4hrs.):** Goals of parallelism (concurrency, throughput), Amdahl's law, Instruction level parallelism (Pipelining, Super scaling (basic features)), Processor level parallelism, Shared memory & Distributed memory features
Teaching / Learning Methods:	Lectures, Tutorials, class discussions, take home exercises, Guided learning
Assessment Methods:	MCQ, Structured Question, Presentation
Assessment Strategy:	Continuous Assessments - 35 % End-Semester Examination - 65 % Final Marks = Continuous Assessment + End-Semester Examination
Recommended Reading(s):	1. Computer Architecture & Organization by Mano, Prentice-Hall 2. Structured Computer Organization by A.S. Tenenbaum, Prentice Hall 3. Computer Organization & Architecture: Designing for performance by W.Stallings, Prentice Hall 4. Computer Architecture & Organization by J.P.Hayes, McGraw-Hill

Course Code	CO1226	**Course Name**	Social Harmony		
Year	I	**Hourly Breakdown**	**Theory**	**Practical**	**Independent Learning**
Semester	II		30	-	70
Core/Optional	Core	**GPA/NGPA**	GPA		
Aim(s) / Objective(s):	Objective of the course is to help students understand peace and conflict in the contemporary situation and the role of communication in conflict scenarios and to help students analyse conflict situations from many different perspectives.				
Intended Learning Outcomes (ILOs):	At the end of the course, students should be able to: - Describe and apply the concepts of conflict and peace and the role of communication in conflict situations.				
Course Content:	**Introduction to Social Harmony (03 Hours):** What is Social Harmony?, Importance, Scope & Nature of Social Harmony, Theories of Harmony, Approaches to Social Harmony; **Element that creates Social Disharmony (05 Hours):** Violence, Substance Abuse, Gender Inequality, Discrimination, Segregation, Ethnicity, Culture and Society, Religious Conservatism & Fanaticism, Economic Inequality; **Historical Factors Historical Background to Social Disharmony (04 hours):** Colonization /Neo-Colonization, Exploitation, Expansion of Religions, Ethnic Cleansing; **Historical Background to Promotion of Social Harmony (05 hours): Institution Level** (United Nations. - NATO, SAARC, ASEAN, ICRC, WCC & Roman Catholic Pontifical Councils, Amnesty International, Human Rights Watch, European Union, Common Wealth of Nations), **Individual Level** (Vivekanandar, Gandi, Vipulanandar, Martin Luther King, Pope john Paul II, Mother Theresa, Nelson Mandela, Umar Rali- 2nd Kalif); **Role of World Religions in the Promotion of Social Harmony (04 Hours):** Hinduism, Buddhism, Christianity, Islam, Inter Religious Dialogue among World Religions, **Education on Human Values (04 Hours):** Tolerance, Compassion, Sympathy, Honesty, Friendship, Kindness, Helping Attitude, Gratitude, Acceptance of Pluralism, Trust, Forgiveness, Empathy; **Conflict Resolution & Peace Building (05 Hours): Conflict Resolution** (Origin & Nature of Conflict, Theories of Conflict, Prevent of Conflict, Mediation in Resolving Conflict, Resolution of Conflict), **Peace Building** (Concept of Peace, Approaches to Peace, Strategies on Establishing Peace, Peace Making/ Peace Building, Peace Keeping, Ethics of Peace)				
Teaching / Learning Methods:	Lectures, Tutorials, class discussions				
Assessment Methods:	MCQ, Structured Question, Presentation				
Assessment Strategy:	Continuous Assessments - 35 % End-Semester Examination - 65 % Final Marks = Continuous Assessment + End-Semester Examination				
Recommended Reading(s):	1. Apfel, L.J. (2011), The Advent of Pluralism: Diversity and Conflict in the Age of Sophocles. 2. Kandiah, Thiru, ed. (2001), The media and the Ethnic Conflict", Colombo:Marga Institute. 3. Fernando, L. (2010), Promoting Ethnic Cohesion in Universities: Possible Activities.				

Year II Semester I

Course Code	CO2121	**Course Name**	Advanced Mathematics for Computing		
Year	II	**Hourly Breakdown**	**Theory**	**Practical**	**Independent Learning**
Semester	I		30	-	70
Core/Optional	Core	**GPA/NGPA**	GPA		
Aim(s) / Objective(s):	This course is designed to provide students with advanced concepts that can be converted to algorithms.				

Intended Learning Outcomes (ILOs):	At the end of the course, students will be able to: - describe matrices and can write algorithms for various matrix operations - interpret the algorithms for the linear 2D and 3D transformations - define vector operations and convert them to suitable algorithms - design suitable algorithms for problems that can use differentiation and integration techniques - implement algorithms for various numerical methods
Course Content:	**Matrices:** Definition of Matrix, Column and Row matrices (vectors), Identity, Null, Diagonal, square matrix, Equal matrices, Matrix addition, scalar multiplication of a matrix, Multiplication of two matrices and properties, Determinants (Cofactor matrix, Computing determinants, Properties of determinants), Singular and Non singular matrices, Inversion of a matrix and properties, Transpose and ad joint of a matrix and their properties, Symmetric, Skew-Symmetric, triangular matrices, System of linear equations; **Linear Transformations:** Two dimensional transformations (Definition and matrix representation of 2D LT, Translation (non LT), Image points and Invariant points under a LT, Image of a straight line under LT, Image of a polygon under LT, Basic trigonometric identities relative to geometric LTs, Rotation, Reflection, Scaling, Shearing parallel to X-axis or Y-axis, Composite LT, Inverse of a LT), Three dimensional transformation (Definition and matrix representation of a 3D LT, Rotation about any line through the origin, Reflection in any plane through the origin, Scaling about the origin); **Vectors:** Definition of Vector and a Scalar, Equality of vectors, Geometric representation of a vector, Modulus of a vector, Unit vector, null vector, constant vector, Multiplication of a vector by a scalar, Addition of vectors and subtraction, Position vectors, Ratio theorem and related topics, Resolution of vector, Base vectors, Scalar product and vector product and their properties, Vector equation of a straight line and plane and the basic problems behind them; **Differentiation and Integration:** Uniform rate of change, Variable rate of change (Average rate, Instantaneous rate), Definition of differentiation, Properties and examples, Higher order derivatives, Integration (Integration as the inverse of differentiation, Integration of standard functions, Integration as area under a curve), Numerical Methods and its algorithmic implementations
Teaching / Learning Methods:	Use of chalkboard, tutorial, textbook assignments, Guided learning, Powerpoint slides
Assessment Methods:	Structured Questions, Group activity
Assessment Strategy:	Continuous Assessments - 35 % End-Semester Examination - 65 % Final Marks = Continuous Assessment + End-Semester Examination
Recommended Reading(s):	1. Business Mathematics by: Qazi Zameeruddin, V.K.Khanna and S.K.Bhambri 2. Higher Algebra by H.S. Hall and Knight 3. An In- Depth Study of Mathematics by Dr. A.B. Mathur

Course Code	CO2122	**Course Name**	Operating System		
Year	II	**Hourly Breakdown**	**Theory**	**Practical**	**Independent Learning**
Semester	I		30	-	70
Core/Optional	Core	**GPA/NGPA**	GPA		
Aim(s) / Objective(s):	This course is designed to introduce various Operating systems and their fundamental issues to the students.				
Intended Learning Outcomes (ILOs):	At the end of the course, students will be able to: - illustrate various operating systems - analyse system structures - distinguish process and threads - design CPU scheduling				

	- illustrate Concurrency problems - interpret deadlock and starvation problems
Course Content:	**Computer system overview:** Basic elements, Processor Registers, Instruction execution, Interrupts, Memory hierarchy, Cache memory, I/O communication techniques, Computer system structures, network structures, LAN, WAN, Wireless LANs, Client-server, Peer to Peer computing, web based and embedded computing, Multi processor & distributed systems (cluster, handheld systems); **Operating system overview:** What is operating system?, OS objectives and functions, The evolution of OS (1, 2, 3, 4 generations), Mainframe OS, Server OS, Multiprocessor OS. Desktop system, Real time OS, Clustered system, Handheld system, Smart card system, Embedded OS, Developments leading to modem OS, MS window overview, Traditional Unix system, Linux; **Operating system structures:** System components, OS Servers, System Calls, System Programs, System Structure, Virtual Machines, System design & implementation, system generation, System boot; **Process management:** Process: (Process concept, process model, process creation, process termination, process hierarchies, process states, process scheduling, operations on processes, cooperating processes, inter process communication, communication in client server system, Unix process management), Threads: (Process & Threads, The thread model, thread usage, implementing threads in user space and kernel, hybrid implementation, scheduler activations, pop up threads, making single - threaded code multi threaded, Windows XP threads and SMP management, Linux threads management, Java threads), CPU scheduling: (Basic Concepts, Scheduling Criteria, scheduling algorithms, multiple processor scheduling, real time scheduling, threaded scheduling, O/S examples, java thread scheduling algorithm evaluation), Concurrency: Mutual exclusion and synchronization (Principles of concurrency, mutual exclusion: Hardware support, Race condition, critical regions, sleep & wake up, semaphores, Mutexes, monitors, message passing, The readers and writers problem, sleeping barber problem, The critical section problem, two - tasks solutions, synchronization H/W, java synchronization, synchronization examples), Concurrency: Dead lock and starvation (Pre-emptable and non pre-emptable resources, resources acquisition, Principles of dead lock, dinning philosophers problem, dead lock prevention, dead lock avoidance bankers, dead lock detection, recovery from dead lock, Linux kernel concurrency mechanism, Condition for dead locks, dead lock modeling, The ostrich algorithm, Dead lock avoidance and recovery (Resource trajectories, safe & unsafe states, Bankers algorithm for single resource, Bankers algorithm for multiple resource), (Dead lock detection with one resource of each type, Dead lock detection with multiple resources of each type, Recovery from dead lock), Dead lock avoidance (Resource trajectories, safe & unsafe states, Bankers algorithm for single resource, Bankers algorithm for multiple resource), Dead lock prevention (Attacking the mutual exclusion condition, Attacking the hold and wait condition, Attacking the no preemption condition, Attacking the circular wait condition), Other uses - two phase locking, non resource dead locks, starvation; **Memory management:** Background, memory partitioning, contiguous memory allocation, paging, segmentation, segmentation with paging, virtual memory (paging, demanding page, page tables, TLB, inverted page table, allocation of frames), Linux and Windows memory management, Page replacement algorithms (Optimal page replacement algorithm, Not recently used page replacement algorithm, First in first out, Second chance page replacement algorithm, the clock page replacement, algorithm, the least recently used and its simulation in software, The working page set page replacement algorithm, The work page set clock page replacement algorithm, Belady's anomaly, stack algorithm, the distance string predicting page, fault rates, Design issues for paging system), Implementation of issues (OS involvement in paging, Page Fault Handling, Instruction backup, Locking pages in memory, backing store, Separation of policy and mechanism), Segmentation (Implementation of pure segmentation, Segmentation with paging: MULTICS, Segmentation with paging: The Intel Pentium); **Scheduling:** Uniprocessor scheduling (Types of processor scheduling, Scheduling algorithms, Unix scheduling), Multi processor and real time scheduling (Multiprocessor scheduling, Real time

	scheduling, Linux scheduling, Windows scheduling); **Input/Output systems:** Overview, I/O H/W, Application I/O interface, Kernal I/O subsystem, Transferring I/O to H/W, Principles of I/O H/W (I/O devices, device controllers, memory mapped I/O, Direct memory access, Interrupt Revisited), Principles of I/O software (Goals of I/O Software, Programmed I/O, Interrupt Driven I/O, I/O using DMA), Mass Storage Structure (Disk Structure, Scheduling, Management, Swap space management, RAID structure, Disk attachment, Stable Storage Implementation), Disks (Disk Hardware, Formatting. Disk Arm Scheduling Algorithm, Error Handling), Clock (Clock Hardware, Clock Software, Soft timers), Character oriented Terminals (RS-232, Terminal Hardware, Input Software, Output Software), Network Management, Power Management; **File management:** Overview, File Concept, Access Methods, File Organization and Access, File Directories, File Sharing, Record Blocking, Protection, Secondary Storage Management, Linux Virtual File System, Windows File System, File System Structure and implementation, Directory implementation, Allocation Method, Free Space Management, Log structured File System, NFS, CD-ROM, CP/M, MS-DOS, Windows 98 File System; **Multimedia operating system:** Multimedia Processors - Hardware, O/S types, Synchronization, Scheduling, Multi computers - Hardware, Low level communication software, User level communication software, Distributed shared memory, Multicomputer scheduling, Load Balancing, Distributed Systems (Netware Hardware, Network services and protocols, Document based middle ware, File system based middle ware, Coordination based middle ware); **Distributed system:** Distributed System structures (Background, Topology, Communication, Protocols, Robustness, Design issues, Example: Networking), Distributed File Systems (Background, Naming and Transparency, Remote File Access, State-full vs State-less services, File Replication, Example: AFS), Distributed Coordination (Event ordering, Mutual Exclusion, Atomicity, concurrency control, Deadlock Handling, Election Algorithm, Reaching Agreement); **Security:** The security environment - threats (Program and System), intruders, Accidental data loss, Basics of cryptography - Secrete key and Public key cryptography, One way functions, Digital Signatures, User Authentication - Using passwords, using a physical object, using Biometrics, Countermeasures, Attack from the inside of the system - Malicious Software: Trojan's Horses, Login Spoofing, Logic Bombs, Trap doors, Buffer Overflow, Generic Security Attacks, Famous Security Flaws, Design Principles for Security, Attacks from outside the system - Virus Damage Scenarios, How virus work and spread, Antivirus and Anti-Antivirus techniques, The Internet worm, Mobile code, Java Security, Protection Mechanisms - Goals of Protection, Domain of Protection, Access matrix, Implementation of Access Matrix, Revocation of Access Rights, Capably based systems, Language Based Protection, Trusted Systems - Trusted Computing Base, Formal Models of Secure Systems, Multilevel security, Orange Book Security
Teaching / Learning Methods:	Use of chalkboard, tutorial, textbook assignments, Guided learning, Powerpoint slides
Assessment Methods:	Structured Questions, Group activity, Group presentation, Multiple Choice Questions, Open book exams
Assessment Strategy:	Continuous Assessments - 35 % End-Semester Examination - 65 % Final Marks = Continuous Assessment + End-Semester Examination
Recommended Reading(s):	1. Modern Operating Systems - By Andrew S.Tanenbaum, Second Edition 2. Operating Systems Internals and Design Principles - By William Stallings, Fifth Edition 3. Operating System Concepts with Java - By Silberschatz, Galvin, Gagne Sixth Edition

Course Code	CO2112	**Course Name**	Practical work on CO2122		
Year	II	**Hourly Breakdown**	**Theory**	**Practical**	**Independent Learning**
Semester	I		-	30	
Core/Optional	Core	**GPA/NGPA**	GPA		

Aim(s) / Objective(s):	This course is designed to teach the students practical implementation of operating system theories.
Intended Learning Outcomes (ILOs):	At the end of the course, students will be able to: - do Linux shell commands - apply Threads using Linux Library - implement Scheduling algorithms and Deadlock Banker Algorithms using C++.
Course Content:	The practical implementation is based on the theory components covered in the course CO2122 Operating System and the lab sessions will be based on the contemporary computer platforms and tools.
Teaching / Learning Methods:	Handouts / Presentations , Laboratory experiments, activities,exercises, Practical records, Tutorial discussion
Assessment Methods:	Individual coding assignment, Classroom and Laboratory assignments, Individual assignments, Laboratory practice
Assessment Strategy:	Continuous Assessments - 35 % End-Semester Examination - 65 % Final Marks = Continuous Assessment + End-Semester Examination
Recommended Reading(s):	1. Modern Operating Systems - By Andrew S.Tanenbaum, Second Edition 2. Operating Systems Internals and Design Principles - By William Stallings, Fifth Edition

Course Code	CO2123	**Course Name**	Software Engineering		
Year	II	**Hourly Breakdown**	**Theory**	**Practical**	**Independent Learning**
Semester	I		30	-	70
Core/Optional	Core	**GPA/NGPA**	GPA		
Aim(s) / Objective(s):	This course is designed to introduce the various development stages of software products and the issues involved until the maintenance phase.				
Intended Learning Outcomes (ILOs):	At the end of the course, students will be able to: - demonstrate software problems - illustrate software development phases - illustrate software requirements and specifications - interpret software design, coding and testing issues - apply maintenance issues - analyse CASE tools and software qualities				
Course Content:	**Introduction (6hrs.):** The Software Problem (Software is expensive, Late, costly, and unreliable, Problem of change and maintenance, Scheduling and quality), Software Products (System software, Application software, Software Product attributes), Software Engineering Approach (The need for an engineering approach, Phased development Process); **Software Process (7hrs.):** Software development process models (Waterfall model, Spiral model, Incremental development), Prototyping (Prototyping techniques, Throw-away Prototyping, Evolutionary Prototyping), Software Project management Process (Phases of management Process, Project management issues); **Software Requirement Analysis and Specification (11hrs.):** Software requirements (Requirement definition, Requirement types), Requirement Analysis (Requirement analysis techniques, Structured analysis, Object-oriented modeling), Requirement specification (Characteristics of software requirement specification (SRS), Components of a SRS, Specification languages), Requirement validation (Requirement reviews, Other methods), Metrics (Size measures, Quality measures); **Software Design (14hrs.):** Design Issues (2hrs.) (Modularity, Abstraction, Encapsulation, Re-usability, Support maintainability), Module-Level concepts (2hrs.) (Coupling, Cohesion), Software design techniques (6hrs.) (Top-down design technique, Bottom-up design techniques, Jackson structured design technique, Data flow oriented design, Object oriented design), User Interface design (2hrs.) (User-system interaction, Information Presentation, User guidance, Interface evaluation), Design				

	Specifications (2hrs.) (Module specification, Structure charts, Class diagrams); **Coding (6hrs.):** Programming practice (3hrs.) (Structured programming, Programming style, Internal documentation), Verification (3hrs.) (Code Inspections, Code reviews, Proving correctness, Symbolic executions); **Testing (7hrs.):** Testing process (2hrs.) (Test plans, Test cases and test criteria, Test case execution and analysis, Test results specification), Testing strategies (2hrs.) (Top-down integration, Bottom-up integration), Testing techniques (3hrs.) (Black-box testing, White-box testing, Alpha testing, Beta testing); **Software Maintenance (4hrs.):** Maintenance types (2hrs.) (Corrective maintenance, Perfective maintenance, Adaptive maintenance), Maintenance process (2hrs.) (Change requests, Impact analysis, System release planning, Change implementation, System release); **Computer Aided Software Engineering (CASE) (2hrs.):** CASE tools (Advantages of using CASE tools, Components of a CASE tool, Function oriented CASE tools - (eg. ORACLE2000), Object oriented CASE tools - (eg. Rational Rose)); **Software Quality (3hrs.):** Software quality processes (Quality reviews, Software standards, Documentation standards, Product quality metrics, Software metrics) ** A Mini Project is to be implemented in developing a software for an office
Teaching / Learning Methods:	Use of chalkboard, tutorial, textbook assignments, Guided learning, Powerpoint slides
Assessment Methods:	Structured Questions, Group activity, Multiple Choice Questions
Assessment Strategy:	Continuous Assessments - 35 % End-Semester Examination - 65 % Final Marks = Continuous Assessment + End-Semester Examination
Recommended Reading(s):	1. Software Engineering by Ian Sommerville, 5th edition, Addison-Wesley, 2000. 2. Software Engineering: A practitioner's approach by Roger S. Pressman, 4th edition, McGraw-Hill International edition, 1997.

Course Code	CO2124	**Course Name**	Internet & Web Design		
Year	II	**Hourly Breakdown.**	**Theory**	**Practical**	**Independent Learning**
Semester	I		30	-	70
Core/Optional	Core	**GPA/NGPA**	GPA		
Aim(s) / Objective(s):	This course is designed to introduce Internet Web Pages and the Server handling of those pages				
Intended Learning Outcomes (ILOs):	At the end of the course, students will be able to: - interpret 4 layer Internet model - illustrate browser concepts and standards - design HTML page design considerations - distinguish client side and server side programming - debate dynamic content techniques and issues in Web Development				
Course Content:	**Introduction to the Internet:** Background and history, The architecture of the Internet, Addressing and naming on the Internet, The basic client-server model of Internet applications, The 4 layered Internet model (The 7 layer OSI model and its 4 layer Internet counterpart, Physical (link), Internet, transport and application layers, Ports, sockets and well-known services); **The World Wide Web:** The idea of hypertext and hypermedia, How the web works - HTTP, HTML and URLs, How the browser works - MIME types, plug-ins and helper applications, The standards - HTML, XML, XHTML and the W3C; **Hypertext Mark - up Language:** The anatomy of an HTML document, Marking up for structure and style (Basic page mark-up, Absolute and relative links, Ordered and unordered lists, Embedding images and controlling appearance (Graphic formats, layouts, borders, effects, Image sizing, thumbnails, colour depth and compression, Image maps), Table creation and use (Call, table and page formatting, As navigation aid, Frames, nesting and targeting)), Descriptive mark-up (Meta tags for common				

	tasks (page refresh & expiration), Semantic tags for aiding search, The Dublin core and RDF), Separating style from structure with style sheets (Internet style specification within HTML, External (linked) style specification using CSS (The object model for cascading sheets, Controlling font size and colour, text and link colours, Background colours, textures and images, Page borders, margins, indents, paragraph and line spacing)), Designing HTML forms (Why forms are needed, Types of information-text areas, buttons, check boxes, radio buttons, Clint side and server side processing of form data), Page and site design considerations (Reducing page weight for enhancing download speed, Optimizing site design for ease of navigation and maintenance); **Client-side Programming:** Introduction (Including scripts in documents: Placing code in an external file, between script tags and in an event handler, Objects, properties, events and methods: Navigation object, Browser object, Document object, Parameters, Methods & functions and Events & Properties), The Java Script syntax (Basic data types, Operators, Control structures, Global functions, Statements), The Java Script object model (Java Script object (Static objects, Core objects: String and Math objects, Data object, Forms object (Submit() and Reset() methods))), Event handling (Events and event handlers, Standard event handler attributes (Mouse related events, Keyboard events, Document events: OnLoading, OnUnloading, OnBlur, OnFocus)), Output in JavaScript (Windows: Window. Alert, Window. Confirm, Document. Write, Window. Prompt, Frames: OnLoad, OnUnload, OnFocus, OnBlur, <Frameset> attribute), Forms handling: Submit(). Reset() Methods, Miscellaneous topics (Cookies, Hidden fields, Images), Applications (Forms handling, Mouse sensitive responses); **Server-side Programming:** Introduction (The need for CGI: Creating dynamic and interactive Web pages, What do we need to run CGI? (The HTTP server, Programming languages for CGI, Configuring the server to support CGI), Some examples: Animations, Client Pull, Push, Access counters, Automatic redirection, Authentication, Executing external programs, Handling forms and other user inputs, Integrating other systems to Web), Input/output operations on the WWW (Passing parameters in and out, Environment variables: Server information and Client information, GET and POST methods, Data encoding and decoding, Response headers, Accept types and Content types, Server redirection), Forms processing (Relevant HTML tags (e.g.FORM), Text and Password fields, Submit and Reset buttons, Radio buttons and Checkboxes, Multi-line text fields, Sending data to the server, Designing applications using forms (Some case studies/Assignments)), Server-side includes (SSI directives, Configuring the server to support SSI, Formatting SSI output), Gateway applications (Sockets, Checking URLS, Databases with flat files, Integrating relational databases using SQL, UNIX manual pages, Email gateway, Search/Index gateway, Image maps), Testing/Debugging CGI applications (Common errors: Directory undefined, Undefined interpret, File permission problems, Malformed header from script, Programming/ system errors, Opening, Closing, Writing files, Problems with environment variables, CGI debugging tools); **Other Dynamic Content Technologies:** Introductions to ASP and JSP, Delivering Multimedia over web pages, The VRML idea, The Java phenomenon - applets and servlets; **Issues in Web Development:** Legal aspects - copyright, Social issues - privacy, PICS, Security concerns - encryption and certification
Teaching / Learning Methods:	Use of chalkboard, tutorial, textbook assignments, Guided learning, Powerpoint slides
Assessment Methods:	Structured Questions, Group activity, Group presentation, Multiple Choice Questions, Open book exams
Assessment Strategy:	Continuous Assessments - 35 % End-Semester Examination - 65 % Final Marks = Continuous Assessment + End-Semester Examination
Recommended Reading(s):	1. Beginning XHTML by Frank Boumpery, Cassandra Greer, Dave Raggettt, Janny Raggett, Sebastian Schnitzenbaumer & Ted Wugofski, WROX press (Indian Shroff Publishes SPD) 1 st edition, ISBN: 81-7366-164-2, 2000

	2. HTML & XHTML: The Definitive Guide by Chuck Musciano, Bill kennedy: 4 th edition, 2000 3. XHTML Black Book by Steven Holzner, 2000 4. Beginning PHP 5 Apache, MySQL, Web Development, 2005, Edition, by Elizabeth

Course Code	CO2114	**Course Name**	Practical work on CO2124		
Year	II	**Hourly Breakdown**	**Theory**	**Practical**	**Independent Learning**
Semester	I		-	30	20
Core/Optional	Core	**GPA/NGPA**	GPA		
Aim(s) / Objective(s):	This course is designed to teach the students practical implementation of web design theories.				
Intended Learning Outcomes (ILOs):	At the end of the course, students will be able to: - design HTML webpage - apply dynamic contents in Web Development				
Course Content:	The practical implementation is based on the theory components covered in the course CO2124 Internet & Web Design and the lab sessions will be based on the contemporary computer platforms and tools.				
Teaching / Learning Methods:	Handouts / Presentations ,Laboratory experiments, activities,exercises, Practical records, Tutorial discussion				
Assessment Methods:	Group /Individual Presentations, Small Projects, Quizzes, Practical assessment tests to solve real world problems				
Assessment Strategy:	Continuous Assessments - 35 % End-Semester Examination - 65 % Final Marks = Continuous Assessment + End-Semester Examination				
Recommended Reading(s):	1. HTML & XHTML: The Definitive Guide by Chuck Musciano, Bill kennedy: 4 th edition, 2000 2. XHTML Black Book by Steven Holzner, 2000 3. Beginning PHP 5 Apache, MySQL, Web Development, 2005, Edition, by Elizabeth				

Course Code	CO2125	**Course Name**	Object Oriented Programming		
Year	II	**Hourly Breakdown**	**Theory**	**Practical**	**Independent Learning**
Semester	I		30	-	70
Core/Optional	Core	**GPA/NGPA**	GPA		
Aim(s) / Objective(s):	This course is designed to introduce object oriented programming using Java Programming Language				
Intended Learning Outcomes (ILOs):	At the end of the course, students will be able to: - apply Object Oriented Language and its merits - test Java based Object Oriented Programs - use Java applets and Java Error Handling - solve Threads - use Input Output Streams and Java Programming Tools				
Course Content:	**Introduction to programming language:** Programming languages: Generation of languages, Translators, Program style and documentation, Basics of Object Oriented Programming and its terminology Evolution, Introduction to Java Programming Language (Java's past, present and future, Java and the Internet), Running Java Programs (Introduction to Java Development Kit (JDK), Creating a source file, Compiling and running the source file, Java byte code file, Java byte code interpreter); **Statements Expressions, Variable and Data types**: Statements and Expressions, Comments in Java (Standard Comments, Document Comments), Literals, Variables (What is variable?, Declaring variables, Notes on variable names), Data types (Primitive Data Types, Reference Data Types); **Operators:** Arithmetic Operators: (+, -, *, /, %, ++, --), Logical Operators: (&&, ||, !), Bitwise Operators: (&, |, ^, <<, >>, >>>, ~), Relational				

	Operators: (==, !=, <, >, <=, >=), Operator Precedence; **Arrays and Control Statements:** One Dimensional and Multidimensional Arrays (Declaring array variables, Creating array objects, Accessing array elements, Changing array elements, Multidimensional arrays), Selective Statements (Selection Statements(if then, if...then...else, Switch, Conditional Operator)), Iterative Statements (For loop, While loop, Do-While, Nested loops), Jump Statements(Break, Continue, Return, Labeled loops); **Objects and Classes:** Definition of a class, Creating and destroying Objects, Defining methods, Parameter Passing: Passing arguments to methods, Constructor Methods: Overloading Constructors, This and super keywords, Recursion: Methods that invoke themselves, Using command line arguments (Passing Arguments to Java Programs Handling Arguments in your Java Program); **Object Oriented Concepts:** Encapsulation (Information Hiding) (Access Modifiers: Controlling access to a class, method, or variable (public, protected, private, package), Other Modifiers: static modifier, abstract modifier, final modifier, synchronized modifier, volatile modifier, native modifier), Polymorphism: (Overloading, Overriding), Inheritance (Inheritance Basis, Overriding Methods, Abstract Classes, Reusability); **Applets:** Applications vs. applets, Creating Applets: Major Applet Activities, Initialization, Starting, Stopping, Destroying, Painting, Passing parameters to Applets, Applet Security; **Error Handling:** Exception Objects, Handling Exceptions: Protecting code and Catching Exceptions try catch clause, finally clause, Throwing Exceptions: throws clause, Defining and generating exception; **Multithreading:** Creating and using threads, Thread Synchronization, Thread Scheduling; **Input and Output:** Streams: Input Stream, Output Stream, Byte Stream and Data Stream, Random Access File Stream, Other Major streams; **Programming Tools:** Overview of JDK tools: The runtime interpreter, The compiler, The applet viewer, The debugger, The class file disassemble, The header and stub file generator, the document generator, Visual Development Tools
Teaching / Learning Methods:	Use of chalkboard, tutorial, textbook assignments, Powerpoint slides, Supervised study, Tutorial discussions
Assessment Methods:	Structured Questions, Group activity, Multiple Choice Questions
Assessment Strategy:	Continuous Assessments - 35 % End-Semester Examination - 65 % Final Marks = Continuous Assessment + End-Semester Examination
Recommended Reading(s):	1. Teach Yourself Java in 21 Days: By Laura Lemay 2. The Java Handbook: By Scott Mervealy 3. The Complete Reference JAVA: By Patrick Naughton and Herbert Schildt 4. Java 2 from Scratch: By Steven Hains, Prentice-Hall 5. Java Unleashed: By Sams Net Publishing

Course Code	CO2115	**Course Name**	Practical work on CO2125		
Year	II	**Hourly Breakdown**	**Theory**	**Practical**	**Independent Learning**
Semester	I		-	30	20
Core/Optional	Core	**GPA/NGPA**	GPA		
Aim(s) / Objective(s):	This course is designed to teach the students practical implementation of object oriented programming.				
Intended Learning Outcomes (ILOs):	At the end of the course, students will be able to: - Build and manipulate with classes using objects. - implement the inheritance and polymorphism concepts - emphasise the importance of abstraction and the reuse of java programs - apply object–oriented concepts and software development tools				
Course Content:	The practical implementation is based on the theory components covered in the course CO2125 Object Oriented Programming (Java) and the lab sessions will be based on the contemporary computer platforms and tools.				

Teaching / Learning Methods:	Handouts / Presentations , Laboratory experiments, activities,exercises, Practical records, Tutorial discussion
Assessment Methods:	Individual coding assignment, Classroom and Laboratory assignments, Individual assignments., Laboratory practice
Assessment Strategy:	Continuous Assessments - 35 % End-Semester Examination - 65 % Final Marks = Continuous Assessment + End-Semester Examination
Recommended Reading(s):	1. Teach Yourself Java in 21 Days: By Laura Lemay 2. The Java Handbook: By Scott Mervealy 3. The Complete Reference JAVA: By Patrick Naughton and Herbert Schildt 4. Java 2 from Scratch: By Steven Hains, Prentice-Hall 5. Java Unleashed: By Sams Net Publishing

Course Code	CO2126	**Course Name**	Sri Lankan Studies		
Year	II	**Hourly Breakdown**	**Theory**	**Practical**	**Independent Learning**
Semester	I		30	-	70
Core/Optional	Core	**GPA/NGPA**	GPA		
Aim(s) / Objective(s):	This course is designed to provide students with knowledge of aspects of Sri Lanka relating to its position in the modern world. It will look in a very general way at areas related to the social sciences in a manner that crosses disciplines.				
Intended Learning Outcomes (ILOs):	Students will be able to illustrate the basic concepts, key terms and areas of Geography, Political Science, Economy and Sociology of Sri Lanka.				
Course Content:	**Constitutional Development in Sri Lanka:** Students will examine the development of representative institutions from British times until the present day. They will also be introduced to the concepts of rights and the different types of rights and the manner in which they can be enforced; **Sri Lanka and its neighbours:** Students will learn about the geographical positions of Sri Lanka and the distinctive features of its topography and landscape. They will also be introduced to the history of South Asian Countries since independence, with particular reference to relations between the countries; **Economic Change:** Students will be made aware of changes that took place in the Sri Lankan economy during the colonial period, with particular reference to the development of a plantation economy. They will also look at present day changes in the context of the global economy, and the shift from a state centred outlook to private sector expansion. Particular topics to be covered will include (Small farmer subsistence & plantation crop section of Sri Lanka (Small farmer in the plantation sector, Paddy sector and the small farmers Economic development & plantation sector), Privatisation in Sri Lanka (Historical Background, Stages of privatisation, Major problems & issues of privatisation), Global Economy (Globalization & Market Economy, Major issues of Globalization)); **International Organizations:** Students will learn about international organisations that affect Sri Lanka. In addition to the United Nations, they will be introduced to the World Bank, the Asian Development Bank and IMF, and their current role in Sri Lanka; **Social change in Sri Lanka:** Students will be introduced to modern developments in Sri Lanka with particular reference to the following: (Caste and class, Gender, Education and Employment, Aging of the population)				
Teaching / Learning Methods:	Use of chalkboard, tutorial, textbook assignments, Powerpoint slides				
Assessment Methods:	Structured Questions, Group activity, Multiple Choice Questions				
Assessment Strategy:	Continuous Assessments - 35 % End-Semester Examination - 65 % Final Marks = Continuous Assessment + End-Semester Examination				

Recommended Reading(s):	1. Anderson, Taylor, **Understanding Sociology**, Wadsworth Cengage Publishers, 2007 2. Anderton.A, **Economics**, 5th Edition, Pearson Longman Publications. Dept. of Surveys, **National Atlas,** 1991 3. **Political Theory in Transition**, Edited by, Noel O Sullivan, Rout ledge publications, 2000

Course Code	GEP - III	**Course Name**	General English Proficiency - III		
Year	II	**Hourly Breakdown**	**Theory**	**Practical**	**Independent Learning**
Semester	I		30	-	70
Core/Optional	Core	**GPA/NGPA**	NGPA		
Aim(s) / Objective(s):	To make the students acquire listening and speaking skills for the technological environment.				
Intended Learning Outcomes (ILOs):	At the end of this course, students will be able to - know oral English in formal and informal situations - involve in effective conversation - read different types of reading materials - write without errors				
Course Content:	**UNIT I: Listening:** Listening to informal conversations and participating; **Speaking:** Opening a conversation (greetings, comments on something, weather) - Turn taking - Closing a conversation (excuses, general wish, positive comment, thanks); **Reading** - Developing analytical skills, Deductive and inductive reasoning - Extensive reading; **Writing:** Effective use of SMS for sending short notes and messages - Using 'emoticons as symbols in email messages; **Grammar:** Regular & irregular verbs - Active and passive voice; Vocabulary - Homonyms (e.g. 'can') - Homophones (e.g. 'some', sum'); **E-materials:** Interactive exercise on Grammar and vocabulary - blogging: **Language Lab:** Listening to different types of conversation and answering questions. **UNIT II: Listening:** Listening to situation based dialogues; **Speaking:** Conversation practice in real life situations, asking for directions (using polite expressions), giving directions (using imperative sentences), Purchasing goods from a shop, Discussing various aspects of a film (they have already seen) or a book (they have already read); **Reading:** Reading a short story or an article from newspaper, Critical reading, Comprehension skills; **Writing:** Writing a review / summary of a story / article, Personal letter (Inviting your friend to a function, congratulating someone for his success, thanking one's friend / relatives); **Grammar:** modal verbs, Purpose expressions; **Vocabulary:** Phrasal verbs and their meanings, Using phrasal verbs in sentences; **E-materials:** Interactive exercise on Grammar and vocabulary, Extensive reading activity (reading stories / novels from links), Posting reviews in blogs; **Language Lab:** Dialogues (Fill up exercises), Recording students' dialogues.				
Teaching / Learning Methods:	Direct Interaction, OnLine Resources, Self Study				
Assessment Methods:	Group activity, Written Test				
Assessment Strategy:	Continuous Assessments - 35 % End-Semester Examination - 65 % Final Marks = Continuous Assessment + End-Semester Examination				
Recommended Reading(s):	1. Rutheford, Andrea J. Basic Communication Skills for Technology, New Delhi, Pearson Education, 2001 2. Savarimuttu, J. S. Rohan and G. Petricia Alphine Nirmala. *English Grammar and Usage: An Ideal Companion for Advanced Learners*. Chennai: New Century Book House (NCBH), June 2016. (ISBN 978-81-2343-204-5) (Code No. A3506)				

Year II Semester II

Course Code	CO2221	**Course Name**	Data Communication Systems		
Year	II	**Hourly Breakdown**	**Theory**	**Practical**	**Independent Learning**
Semester	II		30	-	70
Core/Optional	Core	**GPA/NGPA**	GPA		
Aim(s) / Objective(s):	This course is designed to provide the fundamentals of data communication systems.				
Intended Learning Outcomes (ILOs):	At the end of the course, students will be able to: - demonstrate channel effects on data transmission - appraise the fundamentals of digital communication - Illustrate physical layer characterization and data transmission mechanism - demonstrate data coding for the error recovery and compression				
Course Content:	**Fundamentals of digital communications:** Introduction to digital communications (Definitions of terms, Signal propagation, Sine waves, Square waves, Amplitude, Frequency, Phase), Channel Effects on transmission (Frequency spectra and Fourier analysis, Attenuation, Limited Bandwidth, Delay Distortion, Noise, Data rate in Channels (Nyquist's Theorem, Shannon's Theorem)); **Physical layer characterization:** Modulation (Analog Modulation (Amplitude, Frequency, Phase), Digital Modulation (PSK, QPSK, QAM)), Data Encoding (Binary Encoding (RTZ, NRZ), Manchester Encoding, Differential Manchester Encoding), Transmission Media (Twisted Pair Cables, Co-axial Cables, Fiber Optic Cables, Wireless Media), Physical Layer Interfaces (RS 232 / EIA 232), Last Mile Access Technologies, (Wired: xDSL, FTTH, Wireless: GPRS, EDGE, HSPDA); **Data transmission mechanisms:** Communication Modes (Simplex, Half-duplex, Full-duplex), Transmission Modes (Serial Transmission, Parallel Transmission), Synchronization (Asynchronous transmission, Synchronous Transmission), Switched Communications (Delay and error sensitivities of data and real time traffic, Circuit switching (PSTN), Packet Switching (Datagram mode, Virtual Circuit mode, Integrated switching (ISDN))), Type of Services [Ref 1: pg.32-33] / [Ref 2: pg.503] (Connection Oriented Services, Connectionless Services), Flow Control (Stop-and-Wait Protocol, Sliding Window Protocol), Multiplexing [Ref 1: pg.137-143] (Frequency Division Multiplexing, Synchronous Time Division Multiplexing, Statistical Time Division Multiplexing); **Data coding for error recovery and compression:** Transmission Errors, Error Control (Feedback Error Recovery: ARQ, Forward Error Correction), Error Detection and Correction (Simple Parity Check, Block Sum Check, Hamming Codes, Cyclic Redundancy Check), Lossless Data Compression (LZW, Huffman Encoding)				
Teaching / Learning Methods:	Lecture-demonstration, Use of slides, take home exercises, tutorials, in-class activities				
Assessment Methods:	Essay type questions, Multiple choice questions, Structures questions, Oral questions, Quizzes				
Assessment Strategy:	Continuous Assessments - 35 % End-Semester Examination - 65 % Final Marks = Continuous Assessment + End-Semester Examination				
Recommended Reading(s):	1. Tanenbaum Andrew S., Computer Networks, 4th edition 2. Halsall Fred, Data Communications, Computer Networks and OSI, 4th edition (10th Indian reprinting 2005) 3. William Stallings, Data and Computer Communications, 7th Edition (3rd Impression 2007)				

Course Code	CO2222	**Course Name**	Visual Systems Development Tools		
Year	II	**Hourly Breakdown**	**Theory**	**Practical**	**Independent Learning**
Semester	II		30	-	70
Core/Optional	Core	**GPA/NGPA**	GPA		
Aim(s) / Objective(s):	This course is designed introduce the Visual Systems Development Tools				

Intended Learning Outcomes (ILOs):	At the end of the course, students will be able to: - distinguish object oriented Methodologies over Traditional Methodologies - demonstrate the importance of UML and Rational Rose CASE Tool - illustrate use case diagrams and class diagrams - illustrate various relationships and create relationships in Rational Rose - test object interactions and object behaviours - illustrate system architecture and build the iterations
Course Content:	**Review of the Traditional Methodologies (2hrs):** Review of the Traditional Methodologies (Traditional Methodologies, Advantages of Object Oriented Methodologies over Traditional Methodologies), Classes, Objects, Encapsulation, Association, Aggregation, Inheritance, Polymorphism, States and Transitions; **Visual Modeling using Unified Modeling Language (UML) (3hrs):** What is Visual Modelling? (What is a Model?, Importance of Modeling, Object Oriented Modeling), Introduction to Unified Modeling Language (UML) (History of UML, Overview of UML-Capabilities, Usage of UML), Introduction to Rational Rose CASE tool (Introduction - Importance of Rational Rose, Capabilities of Rational Rose Case Tool); **Introduction to Object or Software Development Process (3hrs):** Introduction, Benefits, Phases and Iterations (Inception Stage: Purpose, Outcome, Evaluation Criteria, Elaboration Stage: Purpose, Outcome, Evaluation Criteria, Construction Stage: Purpose, Outcome, Evaluation Criteria, Transition Stage: Purpose, Outcome, Evaluation Criteria); **Creating Use Case Diagrams (4hrs):** Actors and Use Cases (Actors, Use Cases), Use Case Relationships (Types of Relationships, Stereotypes), Use Case Diagrams in Rational Rose (Creating Main Use Case Diagram in Rational Rose, Creating Relationships in Rational Rose, Creating Additional Use Case Diagrams in Rational Rose), Activity Diagrams (Activities, Transitions, Decision Points, Swim lanes); **Identifying Classes Packages and drawing a Class Diagram (4hrs):** State, Behavior and Identity of Objects, Stereotypes and Classes (Classes (Introduction, Identifying Classes, Stereotypes and Classes)), Creating and Documenting Classes in rational Rose, Packages, Drawing a Class Diagram; **Specifying Relationships (3hrs):** The Need of Defining Relationships, Association and Aggregation Relationships (Association Relationships, Aggregation Relationships), Naming Relationships, Role Names, Multiplicity Indicators, Reflexive Relationships, Package Relationships, Inheritance (Introduction, Single Inheritance versus Multiple inheritances) Finding Relationships, Creating Relationships in Rational Rose; **Discovering Object Interactions (4hrs):** Documenting Scenarios using Interaction Diagrams, Types of Interaction Diagrams (Sequence Diagrams, Collaboration Diagrams); **Adding Behavior and Structure (3hrs):** Representing Behavior and Structure, Creating Attributes and documenting them (Style Guides for defining attributes, Documenting Attributes), Creating Operations and Documenting them, Displaying attributes and operations, Association Classes; **Analysing Object Behavior (3hrs):** Modelling Dynamic Behavior, States, State Transitions, Special States (Start and Stop), State Transition Details, State Details; **Checking the Model (3hrs):** Making the Model Homogeneous, Combining Classes, Splitting Classes, Eliminating Classes, Consistency Checking, Scenario Walk-Through, Event Tracing, Documentation Review; **Designing the system architecture (3hrs):** The need for architecture, The “4+1” view of architecture, The logical view, The component view, The process view, The deployment view, The use case view; **Building the iterations (3hrs):** The Iteration Planning Process (Benefits, Goals), Design the User Interface, Adding Design Classes, The Emergence of Patterns, Designing Relationships, Designing Attributes and Operations, Designing for Inheritance, Coding, Testing, and Documenting the Iteration; **Object Oriented Programming (10hrs):** Introduction to C++, Input/output, Variables, Constants, Data types, operators, Identifiers, Declarations and functions, Control structures 13.5. Arrays and structures, Pointers, Implementing object oriented programming concepts using C++ (Classes, Constructors, Destructors, Copy Constructors 13.7.3. New and delete

	operators), Inheritance, Multiple Inheritance using C++, Polymorphism using C++, Aggregation using C++; **A Case Study Using an object Oriented CASE Tool (12hrs)**
Teaching / Learning Methods:	Lecture-demonstration, Use of slides, take home exercises, tutorials, in-class activities
Assessment Methods:	Essay type questions, Multiple choice questions, Structures questions, Oral questions, Quizzes
Assessment Strategy:	Continuous Assessments - 35 % End-Semester Examination - 65 % Final Marks = Continuous Assessment + End-Semester Examination
Recommended Reading(s):	1. "UML User Guide", Grady Booch, James Rumbaugh, Ivar Jacobson, Addison Wesley, 2000. 2. Visual Modelling With Rational Rose 2000 and UML By Terry Quatrani Foreword by Grady Booch, 2000. 3. "UML Reference Guide", James Rumbaugh, Iver Jacobson, Grady Booch, Wesle, 2000. 4. The objector software development process", Ivar Jacobson,grady booch james Rumbaugh Addison Wesley, 1999. 5. The C++programming language,third edition by Bjarne Stroustrup, 2000. 6. UML Distilled by Maxtin Fowler With Kendall Scot, 2000, Second Edition. 7. Sams Teach Yourself "UML" In 24 Hours By Joseph Schmuuller, 2000.
Platform/Tutorials	**Hardware and Software Requirements:** Hardware (Any standard PC (Pentium)); Software (Windows 95/98/200/NT, Rational Rose 2000, C++ Compiler)

Course Code	CO2212	**Course Name**	Practical work on CO2222		
Year	II	**Hourly Breakdown**	**Theory**	**Practical**	**Independent Learning**
Semester	II		-	30	20
Core/Optional	Core	**GPA/NGPA**	GPA		
Aim(s) / Objective(s):	This course is designed to teach the students practical implementation of statistical theories.				
Intended Learning Outcomes (ILOs):	At the end of the course, students will be able to: - design and draw UML diagrams - specify various relationships and create relationships in Rational				
Course Content:	The practical implementation is based on the theory components covered in the course CO2222 Visual Systems Development Tools and the lab sessions will be based on the contemporary computer platforms and tools.				
Teaching / Learning Methods:	Handouts / Presentations , Laboratory experiments, activities,exercises, Practical records, Tutorial discussion				
Assessment Methods:	Group /Individual Presentations, Small Projects, Quizzes, Practical assessment tests to solve real world problems				
Assessment Strategy:	Continuous Assessments - 35 % End-Semester Examination - 65 % Final Marks = Continuous Assessment + End-Semester Examination				
Recommended Reading(s):	1. "UML User Guide", Grady Booch, James Rumbaugh, Ivar Jacobson, Addison Wesley, 2000. 2. Visual Modelling With Rational Rose 2000 and UML By Terry Quatrani Foreword by Grady Booch, 2000. 3. "UML Reference Guide", James Rumbaugh, Iver Jacobson, Grady Booch, Wesle, 2000. 4. The objector software development process", Ivar Jacobson,grady booch james Rumbaugh Addison Wesley, 1999. 5. The C++programming language,third edition by Bjarne Stroustrup, 2000. 6. UML Distilled by Maxtin Fowler With Kendall Scot, 2000, Second Edition. 7. Sams Teach Yourself "UML" In 24 Hours By Joseph Schmuuller, 2000.				

Course Code	CO2223	**Course Name**	Computer Graphics		
Year	II	**Hourly Breakdown**	**Theory**	**Practical**	**Independent Learning**
Semester	II		30	-	70
Core/Optional	Core	**GPA/NGPA**	GPA		
Aim(s) / Objective(s):	This course is designed to introduce fundamental Computer Graphics				
Intended Learning Outcomes (ILOs):	At the end of the course, students will be able to: - analyse the background of Computer Graphics - illustrate 2D graphics primitives and the 2D transformations - illustrate 3D graphics concepts and 3D transformations - analyse visible surface detection methods - appraise illumination Models and Surface Rendering Methods and color models				
Course Content:	**Introduction to Computer Graphics & Graphics Systems:** Computer Graphics, Computer Graphics Application, Colour Representation, Gray scale representation, Colour models and representation, Computer Graphics Hardware, Cathode Ray Tubes (CRTs), Raster Graphics Devices, Vector Graphics Devices, Computer Graphics Software; **Two Dimensional Graphics Primitives:** Points and Lines, Line Drawing Algorithms, DDA Algorithm, Bresenham’s Algorithm, Parallel Line Algorithm, Frame Buffer, Circle Drawing Algorithms, Using Polar Coordinates: Bresenham’s Algorithm, Mid Point Circle Algorithm, Filled Areas Drawing Algorithms, Scan Line Algorithm, Boundary Fill Algorithm; **Two Dimensional Geometric Transformations & Viewing:** Basic Transformations: Translation, Rotation, Scaling, Matrix Representations and Homogeneous Coordinates, Other Transformations: Shear, Reflection, Composite Transformations: Translations, Scaling’s and Rotations: Rotation around an arbitrary point, Scaling around a fixed point, The 2D Viewing Pipeline, Window to Viewport Mapping, Clipping Operations: Point Clipping, Line Clipping - Cohen Sutherland Algorithm, Polygon Clipping - Sutherland Hodgeman Algorithm; **Three Dimensional Graphics Concepts & Object Representation:** Three Dimensional Coordinate System, Three Dimensional Display Methods: Parallel Projection, Perspective Projection, Hidden Lines and surface removal: Z-buffer algorithm, Polygon Surfaces, meshes and tables, Spline Representations: Interpolation and Approximation Splines, Parametric Continuity conditions, Geometric continuity conditions, Cubic Spline Interpolation methods, Natural cubic splines and Hermite Interpolation, Bezier curves and Surfaces: Bezier curves and their properties, Cubic Bezier curves, Bezier Surfaces, B-spline curves an surfaces: B-spline curves, Uniform, periodic B-splines, Cubic, periodic B-splines, Non-uniform B-splines, spline surfaces, NURB curves and surfaces; **Three Dimensional Geometric Transformations & Viewing:** Translation, Scaling, Rotation, Reflections, Composite Transformations, Viewing pipeline, 3D Viewing Coordinates, Projections- Parallel and Perspective General Projection Transformations **Visible-Surface Detection Methods:** Classification of Visible-Surface Detection Algorithms, Back-Face Detection, Depth-Buffer Method, A-buffer Method, Scan Line Method, Depth Sorting Method, BSP Tree Method, Area Subdivision Method, Octree Method, Ray-casting Method **Illumination Models and Surface Rendering Methods:** Light Sources, Ambient Light, Diffuse Reflection, Specular Reflection, Phong Model, Polygon, Rendering Methods, Constant Intensity Shading, Gouraud Shading, Phong Shading, Ray Tracing Methods **Color Models and Color Applications:** Properties of Light, XYZ color model RGB color model CMYK color model, HSV color model, Conversion between HSV and RGB models				
Teaching / Learning Methods:	Lecture-demonstration, Use of slides, take home exercises, tutorials, in-class activities				
Assessment Methods:	Essay type questions, Multiple choice questions, Structures questions, Oral questions, Quizzes				
Assessment Strategy:	Continuous Assessments - 35 %				

	End-Semester Examination - 65 % Final Marks = Continuous Assessment + End-Semester Examination
Recommended Reading(s):	1. Computer Graphics C Version by Donald Hearn and M. Pauline Baker, Second Edition, Pearson Education, 2007 2. Digital Image Processing by Rafael C. Gonzalez and Richard E. Woods, Pearson 2002, Second Edition 3. Computer Graphics Principles and Practice Second Edition, by James 4. D. Foley, Andeies van Dam, Stevan K. Feiner and John F. Hughes, Addison Wesley, 2000 5. Fundamentals of Three-Dimensional Computer Graphics by Alan Watt, Addison-Wesley

Course Code	CO2213	**Course Name**	Practical work on CO2223		
Year	II	**Hourly Breakdown**	**Theory**	**Practical**	**Independent Learning**
Semester	II		-	30	20
Core/Optional	Core	**GPA/NGPA**	GPA		
Aim(s) / Objective(s):	This course is designed to teach the students practical implementation of statistical theories.				
Intended Learning Outcomes (ILOs):	At the end of the course, students will be able to: - design and develop line drawing computer graphics algorithms - design and develop Circle drawing computer graphics algorithms - design and develop 2D transformation computer graphics algorithms				
Course Content:	The practical implementation is based on the theory components covered in the course CO2223 Computer Graphics and the lab sessions will based on the contemporary computer platforms and tools.				
Teaching / Learning Methods:	Handouts / Presentations , Laboratory experiments, activities,exercises, Practical records, Tutorial discussion				
Assessment Methods:	Group /Individual Presentations, Small Projects, Practical assessment tests to solve real world problems				
Assessment Strategy:	Continuous Assessments - 35 % End-Semester Examination - 65 % Final Marks = Continuous Assessment + End-Semester Examination				
Recommended Reading(s):	1. Computer Graphics C Version by Donald Hearn and M. Pauline Baker, Second Edition, Pearson Education, 2007 2. Digital Image Processing by Rafael C. Gonzalez and Richard E. Woods, Pearson 2002, Second Edition 3. Computer Graphics Principles and Practice Second Edition, by James 4. D. Foley, Andeies van Dam, Stevan K. Feiner and John F. Hughes, Addison Wesley, 2000 5. Fundamentals of Three-Dimensional Computer Graphics by Alan Watt, Addison-Wesley				

Course Code	CO2224	**Course Name**	Human Computer Interaction		
Year	II	**Hourly Breakdown**	**Theory**	**Practical**	**Independent Learning**
Semester	II		30	-	70
Core/Optional	Core	**GPA/NGPA**	GPA		
Aim(s) / Objective(s):	The course is designed to introduce the student to the basic concepts of human-computer interaction. It will cover the basic theory and methods that exist in the field.				
Intended Learning Outcomes (ILOs):	At the end of the course, students will be able to: - demonstrate human, computer, interaction paradigms - illustrate software design process, models and theories - analyse groupware implementation				

Course Content:	**Fundamentals:** Human, Computer, Interaction, Paradigms; **Design Process:** Interactive Design Basics, HCI in the software process, Design rules (Usability, standard, guidelines, Golden rules and heuristics, HCI patterns), Implantation support, Evaluation Techniques, Universal Design, User support; **Models and Theories:** Cognitive models, Scio-Organizational issues and stakeholder requirements, Communication and collaboration models, Task Analysis, Dialog notion and design, Models of the system, Modelling rich interaction; **Group ware:** Groupware systems, Computer mediated communications, meeting and decision support systems, shared application and artefacts, Frameworks for groupware, implementing synchronous groupware
Teaching / Learning Methods:	Lecture-demonstration, Use of slides, take home exercises, tutorials, in-class activities
Assessment Methods:	Essay type questions, Multiple choice questions, Structures questions, Oral questions, Quizzes
Assessment Strategy:	Continuous Assessments - 35 % End-Semester Examination - 65 % Final Marks = Continuous Assessment + End-Semester Examination
Recommended Reading(s):	1. Human - Computer Interaction, By Alan Dix,Janet Finalay, Gregory.D Abowd (3rd Edition, 2004), Persian Edition 2. Human Computer Interaction in the New Millennium, by John M. Carroll, (Persian Edition) 3. Engineering the Human Computer interaction, by Andy Downton, McGraw-Hill International (UK) Limited

Course Code	CO2214	**Course Name**	Practical work on CO2224		
Year	II	**Hourly Breakdown**	**Theory**	**Practical**	**Independent Learning**
Semester	II		-	30	20
Core/Optional	Core	**GPA/NGPA**	GPA		
Aim(s) / Objective(s):	This course is designed to teach the students practical implementation of Human Computer Interaction.				
Intended Learning Outcomes (ILOs):	At the end of the course, students will be able to: - understand interactive designing - design interactive user interfaces - create new interaction styles for specific user interfaces				
Course Content:	The practical implementation is based on the theory components covered in the course CO2224 Human Computer Interaction and the lab sessions will be based on the contemporary computer platforms and tools.				
Teaching / Learning Methods:	Lecture-demonstration, Use of slides, take home exercises, tutorials, in-class activities				
Assessment Methods:	Essay type questions, Multiple choice questions, Structures questions, Quizzes				
Assessment Strategy:	Continuous Assessments - 35 % End-Semester Examination - 65 % Final Marks = Continuous Assessment + End-Semester Examination				
Recommended Reading(s):	1. Human - Computer Interaction, By Alan Dix,Janet Finalay, Gregory.D Abowd (3rd Edition, 2004), Persian Edition				

Course Code	CO2225	**Course Name**	Software Management Techniques		
Year	II	**Hourly Breakdown.**	**Theory**	**Practical**	**Independent Learning**
Semester	II		30	-	70
Core/Optional	Core	**GPA/NGPA**	GPA		

Aim(s) / Objective(s):	This course is designed to provide software management techniques.
Intended Learning Outcomes (ILOs):	At the end of the course, students will be able to: - compare software projects - schedule the plan of software projects - demonstrate software cost and quality management - demonstrate human resource, communication, risk and procurement management - illustrate project management process groups
Course Content:	**Introduction to Project Management:** Importance of software project management, what is a project?, Problems with Software Projects, What is Project Management?, Stages of Project, The Feasibility Study, The Cost-benefit Analysis, Planning, Project Execution, Project and Product Life Cycles, The Stakeholder of Project, All parties of project, The Role of Project Manager, Project Management Framework, Software Tools for Project Management; **Project Planning:** Integration Management, What is Integration Management, Project Plan Development, Plan Execution, Scope Management: What is Scope Management?, Methods for Selecting Projects, Project Charter, Scope Statement, Work Breakdown Structure, Stepwise Project Planning, Overview, Main Steps in Project Planning, Use of Software (Microsoft Project) to Assist in Project Planning Activities; **Project Scheduling:** Time Management, Importance of Project Schedules, Schedules and Activities, Sequencing and Scheduling Activity, Project Network Diagrams, Network Planning Models, Duration Estimating and Schedule Development, Critical Path Analysis, Program Evaluation and Review Technique (PERT), Use of Software (Microsoft Project) to Assist in Project Scheduling; **Project Cost Management:** Importance and Principles of Project Cost Management, Resource Planning, Cost Estimating, Types of Cost Estimates, Expert Judgment Estimating by Analogy, COCOMO Model, Cost Budgeting, Cost Control, Use of Software (Microsoft Project) to assist in Cost Management; **Project Quality Management:** Quality of Information Technology Projects, Stages of Software Quality Management, Quality Planning, Quality Assurance, Quality Control, Quality Standards, Tools and Techniques For Quality Control; **Project Human Resources Management:** What is Project Human Resources Management?, Keys to Managing People, Organisational Planning, Issues in Project Staff Acquisition and Team Development, Using Software to Assist in Human Resource Management; **Project Communication Management:** Communications Planning, Information Distribution, Performance Reporting Administrative Closure, Suggestions for Improving Project Communications, Using Software to Assist in Project Communications; **Project Risk Management:** The Importance of Project Risk Management, Common Sources of Risk in IT projects, Risk Identification, Risk Quantification, Risk Response Development and Control, Using Software to Assist in Project Risk Management; **Project Procurement Management:** Importance of Project, Procurement Management, Procurement Planning, Solicitation, Source Selection, Contract Administration, Contract Close-out; **Project Management Process Groups:** Introduction to Project Management Process Groups, Project Initiation, Project Planning, Project Executing, Project Controlling and Configuration Management, Project Closing
Teaching / Learning Methods:	Lecture-demonstration, Use of slides, take home exercises, tutorials, in-class activities
Assessment Methods:	Essay type questions, Multiple choice questions, Structures questions, Quizzes
Assessment Strategy:	Continuous Assessments - 35 % End-Semester Examination - 65 % Final Marks = Continuous Assessment + End-Semester Examination
Recommended Reading(s):	1. “Information Technology Project Management” Kathy Schwalbe, International Student Edition, THOMSON Course Technology, 2003

	2. "Software Project Management" Bob Hughes and Mike Cotterell, Third Edition, Tata McGraw-Hill 3. "Microsoft Office Project 2003 Bible", Elaine Marmel, Wiley Publishing Inc. 4. Basics of Software Project Management, NIIT, Prentice-Hall India, 2004 5. Software Project Management in Practice, Pankaj Jalote, Pearson Education,2002 6. Software Project Management, A Concise Study, S.A. Kelkar, Revised Edition, Prentice Hall India, 2003
Software Requirements	**Software:** Microsoft Project 2003

Course Code	CO2226	**Course Name**	Automata Theory		
Year	II	**Hourly Breakdown**	**Theory**	**Practical**	**Independent Learning**
Semester	II		30	-	70
Core/Optional	Core	**GPA/NGPA**	GPA		
Aim(s) / Objective(s):	This course is designed to provide basic concepts of finite automata theories				
Intended Learning Outcomes (ILOs):	At the end of the course, students will be able to: - Use finite automata expressions - apply algebraic laws for regular expressions - analyse pumping lemma and application properties of regular expressions - illustrate parsing and parsing trees of a grammar - distinguish deterministic finite automaton and non-deterministic automatons - illustrate Turing Machine and its functions				
Course Content:	Basic concepts of finite automata and languages; Finite state automata, regular expressions and regular languages; Algebraic laws for regular expressions ; Equivalence between DFA and NFA: Regular expression and equivalence to FA; Pumping lemma and application properties of regular languages minimization of automata and applications context-free grammars and languages; parsing (or derivation) and parse trees ambiguity of a grammar and language pushdown automaton (PDA); Deterministic finite automaton, non-determinism; Various forms of PDA Equivalence between CFG and PDA; Chmosky normal form of CFG, pumping lemma; Introduction to Turing Machines				
Teaching / Learning Methods:	Lecture-demonstration, Use of slides, take home exercises, tutorials, in-class activities				
Assessment Methods:	Problem sheets, Essay type questions, Multiple choice questions, Structures questions				
Assessment Strategy:	Continuous Assessments - 35 % End-Semester Examination - 65 % Final Marks = Continuous Assessment + End-Semester Examination				
Recommended Reading(s):	1. Dexter C. Kozen, Automata and Computability, Springer, 1999. 2. J. E. Hopcroft, R. Motwani, and J. D. Ullman, Introduction to Automata Theory, Languages and Computation, Second Edition, Addison Wesley, Reading, MA, 2001. 3. M. Davis, R. Sigal, and E. Weyuker, Computability, Complexity and Languages: Fundamentals of Theoretical Computer Science, Second Edition, Academic Press, New York, NY, 1994. 4. J. Hopcroft, R. Motwani, and J. Ullman. Introduction to Automata Theory, Languages, and Computation, 3rd edition, 2006, Addison-Wesley.				

Year III Semester I

Course Code	CS3121	**Course Name**	Logic Programming and Expert Systems		
Year	III	**Hourly Breakdown.**	**Theory**	**Practical**	**Independent Learning**
Semester	I		30	-	70
Core/Optional	Core	**GPA/NGPA**	GPA		
Aim(s) / Objective(s):	This course is designed to introduce logic programming concepts on Expert Systems				
Intended Learning Outcomes (ILOs):	At the end of the course, students will be able to: - describe Prolog Language - code in Prolog Language - explain rule-based programming concepts				
Course Content:	Introduction to Prolog Language; Prolog syntax and semantics; Lists and operations; Programming techniques; Controlling backtracking; Input / Output; Built-in predicates; Programming style; Data structures; Search strategies; Knowledge representation and expert systems; Rule-based programming; Advanced features, techniques, and applications.;				
Teaching / Learning Methods:	Use of chalkboard, tutorial, textbook assignments, Guided learning, Powerpoint slides				
Assessment Methods:	Structured Questions, Group activity, Multiple Choice Questions				
Assessment Strategy:	Continuous Assessments - 35 % End-Semester Examination - 65 % Final Marks = Continuous Assessment + End-Semester Examination				
Recommended Reading(s):	1. Ivan Bratko. Prolog Programming for Artificial Intelligence, third edition, Addison-Wesley, 2001.				

Course Code	CS3111	**Course Name**	Practical work on CS3121		
Year	III	**Hourly Breakdown**	**Theory**	**Practical**	**Independent Learning**
Semester	I		-	30	20
Core/Optional	Core	**GPA/NGPA**	GPA		
Aim(s) / Objective(s):	This course is designed to teach the students practicals on logic programming and knowledge base.				
Intended Learning Outcomes (ILOs):	At the end of the course, students will be able to: - develop logic programs with the significance of language semantics - devise a plan of action to achieve a goal using standard AI methods				
Course Content:	The practical implementation is based on the theory components covered in the course CS3121 Logic Programming and Expert System and the lab sessions will be based on the contemporary computer platforms and tools.				
Teaching / Learning Methods:	Handouts / Presentations, Laboratory experiments, activities,exercises, Practical records, Tutorial discussion				
Assessment Methods:	Group /Individual Presentations, Small Projects, Quizzes, Practical assessment tests to solve real world problems				
Assessment Strategy:	Continuous Assessments - 35 % End-Semester Examination - 65 % Final Marks = Continuous Assessment + End-Semester Examination				
Recommended Reading(s):	1. Ivan Bratko. Prolog Programming for Artificial Intelligence, third edition, Addison-Wesley, 2001.				

Course Code	CS3122	**Course Name**	Advanced Database Management Systems		
Year	III	**Hourly Breakdown**	**Theory**	**Practical**	**Independent Learning**
Semester	I		30	-	70
Core/Optional	Core	**GPA/NGPA**	GPA		
Aim(s) / Objective(s):	This course is designed introduce advanced database management systems concepts				
Intended Learning Outcomes (ILOs):	At the end of the course, students will be able to: - compare different data models - explain database transactions and recovery procedures - determine concurrency controls and database security - explain client server computing, distributed databases and deductive databases - Compare data warehouse and data mining				
Course Content:	**Data Models:** EER model and relationship to the O0 model, Object Oriented Data Model and ODMG standard, Other data model - NIAM, GOOD, ORM; **Query optimization:** Query execution algorithm, Heuristics in Query Execution, Cost Estimation in Query Execution, Semantic Query Optimization; **Database Transactions and Recovery Procedures:** Transaction Processing Concepts, transaction and system concepts, Desirable properties of a transaction, Schedules and recoverability, Serialisability of Schedules, Transaction support in SQL, Recovery techniques, Database backup; **Concurrency control:** Concurrency control techniques, Granularity of data items; **Database security:** Access privileges, Multi-level security, Statistical database security; **Client Server Computing:** Client Server Concepts, 2-Tier and 3-Tier Client Server Systems, Client/ Server Architecture and the Internet, Client/ Database Server Models, Technology Components of Client Server Systems, Application Development in Client Server Systems; **Distributed Databases:** Reliability and Commit protocols, Fragmentation and Distribution, View Integration, Distributed Database Design, Distributed algorithms for data management, Heterogeneous and Federated Database Systems; **Deductive Databases:** Recursive Queues, Prolog/ Data log Notation, Basic inference Mechanism for Logic Programs, Deductive Database Systems, Deductive Object-Oriented Database Systems; **Data Warehousing and Data Mining:** Data warehousing, Data Mining; **Commercial and Research Prototypes:** Parallel database, Multimedia database, Mobile database, Digital libraries, Temporal Database				
Teaching / Learning Methods:	Use of chalkboard, tutorial, textbook assignments, Guided learning, Powerpoint slides				
Assessment Methods:	Structured Questions, Group activity, Multiple Choice Questions				
Assessment Strategy:	Continuous Assessments - 35 % End-Semester Examination - 65 % Final Marks= Continuous Assessment + End-Semester Examination				
Recommended Reading(s):	1. Fundamentals of Database Systems - By Elmasri R, and Navathe S.B - Addition Wesley 2. Database System Concepts- By Silbershatz A, Korth H.F, and Sutharsan S -McGraw Hill International Edition 3. An Introduction to Data Base Systems - By Date C.J - Addition Wesley 4. Modern Database Management - By McFadden R.F, Hoffer Feffery A, and Prescott Mary B -Bejamin - Cummins (Narosa)				

Course Code	CS3112	**Course Name**	Practical work on CS3122		
Year	III	**Hourly Breakdown**	**Theory**	**Practical**	**Independent Learning**
Semester	I		-	30	20
Core/Optional	Core	**GPA/NGPA**	GPA		
Aim(s) / Objective(s):	This course is designed to teach the students practicals based on advanced techniques to manipulate database management systems.				
Intended Learning Outcomes (ILOs):	At the end of the course, students will be able to: - design a database using standard practices and tools - develop advanced queries to handle information retrieval from databases - explain the concepts of transaction process, concurrency control, and recovery mechanisms - discuss new developments in database technologies and the impacts of emerging database standards				
Course Content:	The practical implementation is based on the theory components covered in the course CS3122 Advanced Database Management Systems and the lab sessions will be based on the contemporary computer platforms and tools.				
Teaching / Learning Methods:	Handouts / Presentations , Laboratory experiments, activities,exercises, Practical records, Tutorial discussion				
Assessment Methods:	Group /Individual Presentations, Small Projects, Practical assessment tests to solve real world problems				
Assessment Strategy:	Continuous Assessments - 35 % End-Semester Examination - 65 % Final Marks = Continuous Assessment + End-Semester Examination				
Recommended Reading(s):	1. R. Elmasri and S.B. Navathe, Fundamentals of Database Systems, 7th Ed., Addison-Wesley, 2015. 2. C.J. Date, An Introduction to Database Systems, 8th Ed., Addison-Wesley, 2003. 3. Ramakrishnan and Gehrke, Database Management Systems, 3rd Ed., McGraw-Hill, 2003.				

Course Code	CS3123	**Course Name**	Systems and Network Administration		
Year	III	**Hourly Breakdown.**	**Theory**	**Practical**	**Independent Learning**
Semester	I		30	-	70
Core/Optional	Core	**GPA/NGPA**	GPA		
Aim(s) / Objective(s):	This course is designed to describe the roles and responsibilities of a system and network administrator.				
Intended Learning Outcomes (ILOs):	At the end of the course, students will be able to: - explain host management, process management, maintenance of log files, and sharing files - determine network management, host and network security - explain automating system administration				
Course Content:	**Describe the roles and responsibilities of a system and network administrator**; **Introduction to Operating Systems**; **Host Management:** (Booting and Shutting Down of an Operating System, Installation and configuration of Software, Proprietary Software, Open-Source Software, Installation and configuration of devices and drivers, Super user / Administrator Privileges, User Management, Adding / Removing users, Controlling User Resources, Disk Space Allocation and quotas, Process Management and Monitoring, Scheduling Processes, Killing / Stopping processes, Restarting a Process, Monitoring Process Activity, Maintaining Log Files, File System Repair, Backup and Restoration, Handling Man Pages/ Help System, Kernel Customization, Managing Heterogeneous Systems, File System Sharing (Samba), Printer Sharing (Samba/CUPS), User IDs, Passwords and Authentication (LDAP), Systems Performance Tuning); **Network Management:** (Introduction to Network Administration Approaches, TCP/IP				

	Networking Basics, IP Addressing and Sub-netting, VLAN Principles and Configuration, Routing Concepts, Network Address Translation, Configuring a Linux Box for Networking, LAN and Wireless LAN, Dial-up and Broadband, Configuring a Linux Box as a Router, Configuring a Web Server (Apache), Configuring a DNS Server (BIND), Configuring Mail Transfer Agents (Postfix), Configuring a Proxy Caches (Squid), TCP/IP Troubleshooting: ping, traceroute, ifconfig, netstat, ipconfig, Network Management, SNMP Ver 2 Basic Components, Commands, Management Information Base, RMON (Host and Network Security, Identify security threats and plan for deployment for preventive methods), Security Planning & System Audits, Security standards and Levels (IS0 15408 standard), Password Security); **Access Control and Monitoring: Wrappers; Firewalls:** (Filtering Rules, Detection and Prevention of Denial of Service (DOS) Attacks, Automatic Identification of Configuration Loopholes (Tripwire), Intrusion Detection Systems, Security Information Resources: CERT (Automating System Administration)); **Use appropriate scripting tools to automate system and network administration:** Use of scripting tools, Shell Scripting Perl / Python Scripting, Use of Make Option; PLATFORM The operating system that is used in this module is Linux Operating System.
Teaching / Learning Methods:	Use of chalkboard, tutorial, textbook assignments, Guided learning, Powerpoint slides
Assessment Methods:	Structured Questions, Group activity, Multiple Choice Question
Assessment Strategy:	Continuous Assessments - 35 % End-Semester Examination - 65 % Final Marks = Continuous Assessment + End-Semester Examination
Recommended Reading(s):	1. Mark Burgess, "Principles of Network and System Administration" (2nd Edition), John Wiley and Sons Ltd, 2004. 2. Craig Hunt, "TCP/IP Network Administration" (3rd Edition), O'Reilly and Associates Inc., 2002. 3. Matthias Kalle Dalheimer and Matt Welsh, "Running Linux", (5th Edition), O'Reilly and Associates Inc., 2007. 4. AEleen Frisch, "Essential System Administration", 3rd Edition, O'Reilly and Associates Inc.,2003

Course Code	CS3113	**Course Name**	Practical work on CS3123		
Year	III	**Hourly Breakdown**	**Theory**	**Practical**	**Independent Learning**
Semester	I		-	30	20
Core/Optional	Core	**GPA/NGPA**	GPA		
Aim(s) / Objective(s):	This course is designed to teach the students practicals on networking, configuration and administration.				
Intended Learning Outcomes (ILOs):	At the end of the course, students will be able to: - create subnetworks for Local Area Network - configure routing protocols and manage it				
Course Content:	The practical implementation is based on the theory components covered in the course CS3123 Systems and Network Administration and the lab sessions will be based on the contemporary computer platforms and tools.				
Teaching / Learning Methods:	Handouts / Presentations , Laboratory experiments, activities,exercises, Practical records, Tutorial discussion				
Assessment Methods:	Group /Individual Presentations, Small Projects, Quizzes, Practical assessment tests to solve real world problems				

Assessment Strategy:	Continuous Assessments - 35 % End-Semester Examination - 65 % Final Marks = Continuous Assessment + End-Semester Examination
Recommended Reading(s):	1. Craig Hunt, "TCP/IP Network Administration" (3rd Edition), O'Reilly and Associates Inc., 2002. 2. AEleen Frisch, "Essential System Administration", 3rd Edition, O'Reilly and Associates Inc.,2003

Course Code	CS3124	**Course Name**	Data Security		
Year	III	**Hourly Breakdown.**	**Theory**	**Practical**	**Independent Learning**
Semester	I		30	-	70
Core/Optional	Core	**GPA/NGPA**	GPA		
Aim(s) / Objective(s):	This course is designed to introduce data security concepts				
Intended Learning Outcomes (ILOs):	At the end of the course, students will be able to: - explain cryptography - compare DES and AES - determine public key cryptography, RSA and ECC - explain security services				
Course Content:	Overview of Cryptography and Number Theory. Block Ciphers.; Stream Ciphers. Results from Information Theory.; Data Encryption Standard (DES).; Advanced Encryption Standard (AES); More on Block Ciphers.; Introduction to Public-Key Cryptography; RSA.; Discrete Logarithm (DL) Problem.; Elliptic Curve Cryptography (ECC); ElGamal Encryption Scheme. Digital Signatures.; Hash Functions. Message Authentication Codes (MACs); Security Services, Key Establishment.;				
Teaching / Learning Methods:	Use of chalkboard, tutorial, textbook assignments, Guided learning, Powerpoint slides				
Assessment Methods:	Structured Questions, Group activity, Multiple Choice Questions				
Assessment Strategy:	Continuous Assessments - 35 % End-Semester Examination - 65 % Final Marks = Continuous Assessment + End-Semester Examination				
Recommended Reading(s):	1. Applied Cryptography: Protocols, Algorithms and Secure Code in C. Bruce Schneier. John Wiley & Sons, 2nd Edition. 2. Introduction to Cryptography with Coding Theory. Wade Trappe and Lawrence C. Washington. Prentice Hall, 2nd edition. 3. Computer Security: Principles and Practice, by William Stallings and Lawrie Brown. Published by Pearson/Prentice Hall, © 2008. ISBN: 0-13-600424-5 4. Cryptography: Theory and Practice. Doug Stinson. Chapman & Hall/CRC, 3rd Edition.				

Course Code	CS3114	**Course Name**	Practical work on CS3124		
Year	III	**Hourly Breakdown**	**Theory**	**Practical**	**Independent Learning**
Semester	I		-	30	20
Core/Optional	Core	**GPA/NGPA**	GPA		
Aim(s) / Objective(s):	This course is designed to teach the students practical implementation of Data Security theories.				
Intended Learning Outcomes (ILOs):	At the end of the course, students will be able to: - program different security mechanisms				

	- apply suitable crypto techniques for secure transfer - design new crypto algorithms
Course Content:	The practical implementation is based on the theory components covered in the course CS3124 Data Security and the lab sessions will based on the contemporary computer platforms and tools.
Teaching / Learning Methods:	Handouts / Presentations ,Laboratory experiments, activities,exercises, Practical records, Tutorial discussion
Assessment Methods:	Group /Individual Presentations, Small Projects, Quizzes, Practical assessment tests to solve real world problems
Assessment Strategy:	Continuous Assessments - 35 % End-Semester Examination - 65 % Final Marks = Continuous Assessment + End-Semester Examination
Recommended Reading(s):	1. Applied Cryptography: Protocols, Algorithms and Secure Code in C. Bruce Schneier. John Wiley & Sons, 2nd Edition. 2. Introduction to Cryptography with Coding Theory. Wade Trappe and Lawrence C. Washington. Prentice Hall, 2nd edition.

Course Code	CS3135	**Course Name**	Theory of Computing		
Year	III	**Hourly Breakdown**	**Theory**	**Practical**	**Independent Learning**
Semester	I		45	-	70
Core/Optional	Core	**GPA/NGPA**	GPA		
Aim(s) / Objective(s):	This course is designed to provide theory of computing principles.				
Intended Learning Outcomes (ILOs):	At the end of the course, students will be able to: - explain computability theory - determine computational complexities - explain formal systems in Computer Science - verify on-line algorithms, game theory, social networks, randomization, and quantum computing				
Course Content:	**Overview of proof techniques, finite automata, nondeterminism, regular languages, pushdown automata, context-free languages and grammars; Computability theory:** (Turing machines, recursively enumerable and recursive languages, Church-Turing thesis, Limitations of algorithms: universality, the halting problem and undecidability); **Computational complexity:** (Definition of complexity, Big O /Big Theta notation, NP-Completeness and Cook's theorem); **Formal systems in Computer Science:** (Formal systems, A formal system for propositional logic, Soundness and completeness, Hoare logic for automatic program verification); **Cryptography, on-line algorithms, game theory, social networks, randomization, and quantum computing**				
Teaching / Learning Methods:	Use of chalkboard, tutorial, textbook assignments, Guided learning, Powerpoint slides				
Assessment Methods:	Structured Questions, Group activity, Multiple Choice Questions				
Assessment Strategy:	Continuous Assessments - 35 % End-Semester Examination - 65 % Final Marks = Continuous Assessment + End-Semester Examination				
Recommended Reading(s):	1. Michael Sipser, Introduction to the Theory of Computation (Second Edition). 2. John Hopcroft, Rajeev Motowani, and Jeffrey Ullman, Automata Theory, Languages, and Computation. (Third Edition)				

	3. Thomas Sudkamp, Languages and Machines: An Introduction to the Theory of Computer science. (Third Edition) 4. Sipser M, Introduction to the Theory of Computation, PWS, 1997.

Course Code	EC3101	**Course Name**	Foundations of Management		
Year	III	**Hourly Breakdown**	**Theory**	**Practical**	**Independent Learning**
Semester	I		45	-	70
Core/Optional	Core	**GPA/NGPA**	GPA		
Aim(s) / Objective(s):	This course is designed to introduce fundamental management concepts.				
Intended Learning Outcomes (ILOs):	At the end of the course, students will be able to: - explain management theories and principles - solve managerial problems				
Course Content:	**Introduction of Management** **Organizing** **Motivation** **Leadership** **Communication** **Controlling**				
Teaching / Learning Methods:	Lecture Discussions, Q and A sessions, Self-studies				
Assessment Methods:	Written Test, Report, Group Presentation				
Assessment Strategy:	Continuous Assessments - 35 % End-Semester Examination - 65 % Final Marks = Continuous Assessment + End-Semester Examination				
Recommended Reading(s):	1. Robbins, Stephen P, and Mary Coutler:Management, Prentice Hall, New Delhi. 2. Principles of Management, 2015, 13: 9781946135186, UNIVERSITY OF MINNESOTA LIBRARIES PUBLISHING EDITION, 2015 3. Principles of Management,Tony Morden,2014,2nd edition, ISBN 9781032022505, March 31, 2021 ,Routledge 4. Management: Principles and Practice" by S K Mandal 5. Management" by Stoner J A and Freeman R E				

Year III Semester II

Course Code	CS3221	**Course Name**	Assembly Language Programming		
Year	III	**Hourly Breakdown**	**Theory**	**Practical**	**Independent Learning**
Semester	II		30	-	70
Core/Optional	Core	**GPA/NGPA**	GPA		
Aim(s) / Objective(s):	This course is designed to provide an introduction to assembly languages and programming concepts.				
Intended Learning Outcomes (ILOs):	At the end of the course, students will be able to: - compare the basic difference between high level languages and low level languages - create basic programs in assembly languages according to the internal structure of the computer				
Course Content:	Introduction to assembly languages and computer organization, Simple programs: assembling, linking, running, debugging, Arithmetic flags and operations, Jumps and loops, Structured assembly language programs, Bit operations, Large programs – an extended example:				

	calculator., File manipulation, Device drivers, Addressing modes and encoding, Advanced assembly instructions
Teaching / Learning Methods:	Use of chalkboard, tutorial, textbook assignments, Guided learning, Powerpoint slides
Assessment Methods:	Structured Questions, Group activity, Multiple Choice Questions
Assessment Strategy:	Continuous Assessments - 35 % End-Semester Examination - 65 % Final Marks = Continuous Assessment + End-Semester Examination
Recommended Reading(s):	1. Computer Organization and Assembly Language Programming for IBM PCs and Compatibles, 2nd ed., by Michael Thorne. 2. A86/D86 manual, by Eric Roberts

Course Code	CS3211	**Course Name**	Practical work on CS3221		
Year	III	**Hourly Breakdown**	**Theory**	**Practical**	**Independent Learning**
Semester	II		-	30	20
Core/Optional	Core	**GPA/NGPA**	GPA		
Aim(s) / Objective(s):	This course is designed to teach the students practical implementation of Assembly Language Programming theories.				
Intended Learning Outcomes (ILOs):	At the end of the course, students will be able to: - demonstrate fundamental assembly language programs concepts - solve variety of computational problems - create programs using fundamental concepts of assembly language programs				
Course Content:	The practical implementation is based on the theory components covered in the course CS3221 Assembly Language Programming and the lab sessions will be based on the contemporary computer platforms and tools.				
Teaching / Learning Methods:	Handouts / Presentations , Laboratory experiments, activities, exercises, Practical records, Tutorial discussion				
Assessment Methods:	Group /Individual Presentations, Small Projects, Practical assessment tests to solve real world problems				
Assessment Strategy:	Continuous Assessments - 35 % End-Semester Examination - 65 % Final Marks = Continuous Assessment + End-Semester Examination				
Recommended Reading(s):	1. D. A. Patterson and J. L. Hennessy, Computer Organization and Design: The Hardware and Software Interface, Morgan Kaufmann Publishers, 5th Ed, 2013. 2. M. Abo-El-Barr and H. El-Rewini, Fundamentals of Computer Organization and Architecture, A John Wiley & Sons Publication, 2004. 3. W. Stallings, Computer Organization and Architecture, Prentice Hall Publishers, 10th Ed, 2015.				

Course Code	CS3222	**Course Name**	Software Quality Assurance		
Year	III	**Hourly Breakdown**	**Theory**	**Practical**	**Independent Learning**
Semester	II		30	-	70
Core/Optional	Core	**GPA/NGPA**	GPA		
Aim(s) / Objective(s):	This course is designed to introduce software quality assurance principles				
Intended Learning Outcomes (ILOs):	At the end of the course, students will be able to: - propose various standards on Software Quality Assurance - evaluate software quality metrics - organise future of Software Quality Assurance				

Course Content:	This course introduces concepts, metrics, and models in software quality assurance. The course covers components of software quality assurance systems before, during, and after software development. It presents a framework for software quality assurance and discusses individual components in the framework such as planning, reviews, testing, configuration management, and so on. It also discusses metrics and models for software quality as a product, in process, and in maintenance. The course will include case studies and hands-on experiences. Students will develop an understanding of software quality and approaches to assure software quality. Introduction to Software, Software Quality Factors, Components of SQA, Pre-project components, Defect removal effectiveness, Reviews, Testing, Maintenance and external participants, Configuration management, Standards, Software quality metrics, Cost of software quality, Software reliability models, In-process quality metrics and models, Future of SQA
Teaching / Learning Methods:	Use of chalkboard, tutorial, textbook assignments, Guided learning, Powerpoint slides
Assessment Methods:	Structured Questions, Group activity, Group presentation, Multiple Choice Questions, Open book exams
Assessment Strategy:	Continuous Assessments - 35 % End-Semester Examination - 65 % Final Marks = Continuous Assessment + End-Semester Examination
Recommended Reading(s):	1. Software Quality Assurance: From Theory to Implementation, by Daniel Galin, Addison Wesley, 2003 2. Metrics and Models in Software Quality Engineering (2nd Edition) by Stephen Kan

Course Code	CS3212	**Course Name**	Practical work on CS3222		
Year	III	**Hourly Breakdown**	**Theory**	**Practical**	**Independent Learning**
Semester	II		-	30	20
Core/Optional	Core	**GPA/NGPA**	GPA		
Aim(s) / Objective(s):	This course is designed to teach the students practicals on testing tools used for Software Quality.				
Intended Learning Outcomes (ILOs):	At the end of the course, students will be able to: - use different automation tools - identify programming bugs using testing tools				
Course Content:	The practical implementation is based on the theory components covered in the course CS3222 Software Quality Assurance and the lab sessions will be based on the contemporary computer platforms and tools.				
Teaching / Learning Methods:	Handouts / Presentations ,Laboratory experiments, activities, exercises, Practical records, Tutorial discussion				
Assessment Methods:	Group /Individual Presentations, Small Projects, Practical assessment tests to solve real world problems				
Assessment Strategy:	Continuous Assessments - 35 % End-Semester Examination - 65 % Final Marks = Continuous Assessment + End-Semester Examination				
Recommended Reading(s):	1. Software Quality Assurance: From Theory to Implementation, by Daniel Galin, Addison Wesley, 2003				

Course Code	CS3233	**Course Name**	Professional Issues in IT		
Year	III	**Hourly Breakdown**	**Theory**	**Practical**	**Independent Learning**
Semester	II		45	-	105
Core/Optional	Core	**GPA/NGPA**	GPA		

Aim(s) / Objective(s):	This course is designed to introduce Professional Issues in IT
Intended Learning Outcomes (ILOs):	At the end of the course, students will be able to: - consider law and government policies regarding issues in IT - evaluate nature of software professionals - propose professional bodies in computing
Course Content:	**Law and Government :** Describe what the law is, Explain the difference between criminal law and civil law, Define the terms legislature, judiciary and executive and outline how these are implemented in different countries with respect to unitary and federal states (e.g. : UK, Sri Lanka, USA)., **The Nature of a Profession:** Outline the characteristics of a profession, Explain what a professional body is, Describe how a professional body is set up and what their main functions are, Discuss your views about the compulsory registration of Software Engineers., **Professional Bodies in Computing:** Outline the most important professional bodies in computing in the world, Describe how these professional bodies serve their members and the public, Describe the obligations that professional bodies in computing impose on, their members and be familiar with the code of conduct of BCS, List the membership categories of some of these professional bodies and how they are awarded
Teaching / Learning Methods:	Use of chalkboard, tutorial, textbook assignments, Guided learning, Power point slides
Assessment Methods:	Structured Questions, Group activity, Group presentation, Multiple Choice Questions, Open book exams
Assessment Strategy:	Continuous Assessments - 35 % End-Semester Examination - 65 % Final Marks = Continuous Assessment + End Semester Examination
Recommended Reading(s):	1. "Professional Issues in Information Technology" by Frank Bott, First south Asia Edition. Chennai Micro Print (P) Ltd., Chennai, India. 2007 (ISBN 1-902505-65-4) 2. BCS Code of conduct: http://www.bcs.org/server.php?show=nav.6030 3. BCS Code of Practice: http://www.bcs.org/server.php?show=nav.6029 4. ACS Code of Ethics: http://courses.cs.vt.edu/~cs3604/lib/WorldCodes/Australia.Code.html 5. ACS Code of Professional Conduct and Professional Practice: http://www.acs.org.au/index.cfm?action=show&conID=copc 6. IEEE Code of Ethics: http://www.iece.org/portal/pages/iportals/aboutus/ethics/code.html 7. Computer Society of Sri Lanka (CSSL): http://www.cssl.lk/index.php?option=com_frontpage&Itemid=66

Course Code	CS3224	**Course Name**	Computer Networks		
Year	III	**Hourly Breakdown**	**Theory**	**Practical**	**Independent Learning**
Semester	II		30	-	70
Core/Optional	Core	**GPA/NGPA**	GPA		
Aim(s) / Objective(s):	This course is designed to introduce various Computer Networks and its Protocols				
Intended Learning Outcomes (ILOs):	At the end of the course, students will be able to: - create network architectures - organise network protocols - measure network performance				
Course Content:	**Network Architectures :** Introduction to Computer Networks, Network Topologies: Bus, Star, Ring; Types of Networks: Local Area Networks, Wide Area Networks, Personal Area Networks; Layered Network Model: OSI model, TCP/ IP model; **Internet Protocols:** Introduction: History of Internet Protocols, Internet Protocol stack; IP Addressing and Routing (Version 4), IP address				

	classes / CIDR; Sub netting: Fixed and variable length; Unicast routing algorithms: RIP, OSPF and IP multi casting; Transport Layer protocols: TCP, UDP; IP Support Protocols: ARP, DHCP, ICMP; Application Layer Protocols: Domain Name System (DNS); Email - SMTP, POP, IMAP; FTP; HTTP: RTP and Vo IP; Overview of IP version 6; **Local Area Networks:** LAN Architectures: Channel Access Methods: Aloha, CSMA, CSMA/CD, CSMA/CA,MACA, CDMA, Token Passing; IEEE 802 standards: 802.3, 802.11, 802.15; Switch Ethernet: Fast Ethernet, Gigabit Ethernet, 10Gb Ethernet; Wireless LANs: 802.11; Frequency Bands (ISM); Operating Modes (adhoc Managed); Variants: 802.11 a/ b/ g/ n; LAN interconnecting devices: Hubs, L2/L3 Switch, Wireless Access Point, Router; **Introduction to Network Monitoring and Management:** Remote Monitoring Techniques: Polling, Traps, SNMP and MIBs; Security management: Firewalls and NAT, VLANs, VPNs; Proxy Servers; Wireless security; Performance Management: Quality of Service over IP, Service Level Managements
Teaching / Learning Methods:	Use of chalkboard, tutorial, textbook assignments, Guided learning, Powerpoint slides
Assessment Methods:	Structured Questions, Group activity, Group presentation, Multiple Choice Questions, Open book exams
Assessment Strategy:	Continuous Assessments - 35 % End-Semester Examination - 65 % Final Marks = Continuous Assessment + End-Semester Examination
Recommended Reading(s):	1. Tanenbaum Andrew S., Computer Networks, 4th edition(2nd Impression 2006) 2. Comer Douglas E, Internetworking with TCP/IP, Volume 1-Principles, Protocols and Architecture, 4th edition, 2002, Prentice-Hall

<table>
<tr><td><b>Course Code</b></td><td>CS3235</td><td><b>Course Name</b></td><td colspan="3">Industrial Training/Project</td></tr>
<tr><td><b>Year</b></td><td>III</td><td rowspan="2"><b>Hourly Breakdown</b>.</td><td><b>Theory</b></td><td><b>Practical</b></td><td><b>Independent Learning</b></td></tr>
<tr><td><b>Semester</b></td><td>II</td><td colspan="3">300 hrs.</td></tr>
<tr><td><b>Core/Optional</b></td><td>Core</td><td><b>GPA/NGPA</b></td><td colspan="3">GPA</td></tr>
<tr><td><b>Aim(s) / Objective(s):</b></td><td colspan="5">To produce the knowledgeable, skilled and experienced graduates, demanded by employers, who are able to apply the knowledge acquired at university to the working world.</td></tr>
<tr><td><b>Intended Learning Outcomes (ILOs):</b></td><td colspan="5">At the successful completion of the training, students will be able to:
- Identify the expected software engineering responsibilities and ethics of work
- Integrate knowledge acquired from academic courses to industrial environment
- Implement and exchange knowledge and skills needed in software engineering projects</td></tr>
<tr><td><b>Note:</b></td><td colspan="5">Industrial Training/Project is a mandatory course for all the students and carries 3 GPA credits as this course is a training programme at the software industries for 6 months. Therefore, as soon as the semester examinations are over, they will be placed at the software industries to fulfil the Industrial Training requirement.</td></tr>
<tr><td><b>Assessment Strategy:</b></td><td colspan="5"><table>
<tr><td><b>Type</b></td><td><b>Marks</b></td></tr>
<tr><td>Final Viva- Voce Examination</td><td>40%</td></tr>
<tr><td>Final Report</td><td>60%</td></tr>
<tr><td><b>Total</b></td><td><b>100%</b></td></tr>
</table>
Both the report and viva-voce examination are mandatory. Students must obtain a minimum of 50 % in each component to successfully complete the industrial training.</td></tr>
</table>

<table>
<tr><td><b>Course Code</b></td><td></td><td><b>Course Name</b></td><td colspan="3">Research Work</td></tr>
<tr><td><b>Year</b></td><td>III</td><td rowspan="2"><b>Hourly Breakdown.</b></td><td><b>Theory</b></td><td><b>Practical</b></td><td><b>Independent Learning</b></td></tr>
<tr><td><b>Semester</b></td><td>II</td><td>-</td><td>300 hrs.</td><td>-</td></tr>
<tr><td><b>Core/Optional</b></td><td>Core</td><td><b>GPA/NGPA</b></td><td colspan="3">NGPA</td></tr>
<tr><td><b>Aim(s) / Objective(s):</b></td><td colspan="5">To engage students in research activities in the field of computer science as per their interested area.</td></tr>
<tr><td><b>Intended Learning Outcomes (ILOs):</b></td><td colspan="5">At the end of the course, students will be able to:
- Study and examine emerging topics in computer science
- Reorganise and evaluate a project
- Write and interpret the research activities
- Produce research findings</td></tr>
<tr><td><b>Note:</b></td><td colspan="5">Research Work is a mandatory course unit for all students, where students should be involved in individual research work in the field of computer science. Students should identify their own research interests and submit the proposal. The head of the department assigns a supervisor in the relevant field.</td></tr>
<tr><td><b>Assessment Strategy:</b></td><td colspan="5">End-of-course Assessment only
<table>
<tr><td><b>Type</b></td><td><b>Marks</b></td></tr>
<tr><td>Final Presentation</td><td>20%</td></tr>
<tr><td>Final Viva- Voce Examination</td><td>20%</td></tr>
<tr><td>Dissertation</td><td>60%</td></tr>
<tr><td><b>Total</b></td><td><b>100%</b></td></tr>
</table>
Research dissertation, presentation and viva-voce examinations are mandatory. Students must obtain a minimum of 50% in each component to successfully complete the research work.</td></tr>
</table>

10. FACULTY CAREER GUIDANCE CELL (FCGC)

FCGC of FAS is being a centre for preparing FAS undergraduates to choose an optimal career through proper industrial exposure. FCGC guides and facilitates FAS undergraduates to make the right career choices and improve their soft skills to successfully manage their academic, personal, and social lives.

10.1 MAIN FUNCTION

- Provide career guidance to all students from the first year onwards to focus on their future careers.
- Organize and conduct career skill development workshops / seminars / personal career mentoring.
- Provide assistance to choose and proceed on an optimal career path, based on students' ability, desire and available opportunities.
- Developing networks for future career
- Awareness workshop on Career pathway in working environment
- Industrial Visits
- Integrating Yoga into Career Counselling

11. FACULTY QUALITY ASSURANCE CELL (FQAC)

The Faculty Quality Assurance Cell (FQAC) serves as a basis for upholding and advancing academic excellence at the Faculty of Applied Science (FAS), Trincomalee Campus, EUSL. FQAC/FAS is striving to elevate the quality of degree programmes through regular assessment, feedback mechanisms, and collaborative efforts.

The Faculty Quality Assurance Cell (FQAC) of the Faculty of Applied Science, Trincomalee Campus, Eastern University, was established in 2015 according to the guidelines given by the Quality Assurance Council of UGC. FQAC has a broad mandate of coordinating all the quality assurance related activities within the faculty under the guidance with the Quality Assurance Unit (QAU) of the Eastern University in line with the UGC circular 04/2015.

11.1 RESPONSIBILITIES

The FQAC/IQAC of a Faculty of Study is responsible for developing the guidelines relating to the quality assurance activities of the faculty and overseeing the implementation of such activities.

These activities will revolve around the following aspects.

1. Curriculum development, management and review
2. Teaching, learning and assessment methods
3. Learning environment (learning opportunities, resources and locations)
4. Academic staff (staff training, upgrading knowledge and skills, student and peer observation, reflection etc.)
5. Administrative staff (general administration and documentation)
6. Student support services (including academic guidance and counselling)
7. Students (including student progress and their achievements)