Department of Computer Science

Faculty of Science

University of Jaffna, Sri Lanka

BSc Hons (Computer Science)

Study Programme Prospectus

(For Direct-intake Computer Science Students)

Academic Year: 2020/2021

October 2022

CONTENTS

INTRODUCTION

University of Jaffna was founded in 1974 as the "Jaffna Campus" of the University of Sri Lanka, and then in 1979, it became an independent autonomous national University in the northern region of Sri Lanka. With the vision of being a leading centre for quality education and high-end research in Information and Communication technology, the Department of Computer Science (DCS) was established in 1991 in the Faculty of Science, University of Jaffna. Since then, the DCS contributes at its best to the development of higher education in the nation.

The Computer Science study programmes at the DCS have a good track record for more than two decades for their curricula and the quality graduates they produce, and hence, there is a very high demand for its degree programmes among the students. The DCS receives the services of ten permanent academic staff, two permanent academic support staff, and four non-academic staff. Students from all parts of Sri Lanka, irrespective of race and religion, join hands with the DCS to meet their academic needs, and add a reputation to the department as well as the university. The student population of mixed ethnic and religious groups provides a better opportunity for students to get familiarised with different cultures through collaborative activities organised by the student societies and leads to a harmonious country.

Since Computer Science is an ever-evolving discipline, in order to cater the changing needs of the contemporary world, the syllabi of the degree programmes have been regularly revised as per ACM/IEEE guidelines in consultation with experts from reputed universities.

The DCS maintains a good industry linkage with leading IT companies in the country, and industrial training is given to all students as part of the curriculum. In addition, Industrial Visits, TechTalks, Coding Competitions, and Workshops are organized by DCS to make students aware of new technologies.

For career development, CS-Career Fair and Life-skill programmes are organised every year. Outreach activities are organised to produce socially responsible graduates. Cultural and religious events are conducted to widen the social and cultural dimensions of the educational experience. Many students attend international conferences and publish full papers based on their final year research projects with the support of their supervisors, and improve their networking and social skills. Furthermore, students perform several activities through the Computer Society (CompSoc) of the University of Jaffna and the IEEE Student Branch of the University of Jaffna, such as, organizing workshops, tech-talks, webinars, etc. In addition to that the CompSoc releases a technical newsletter named as "கணனியம்" (Kananiyam), and conducts seminars to G.C.E. (O/L) and G.C.E. (A/L) students, on ICT subjects and past examination papers.

DEGREE PROGRAMMES

From the date of establishment, since 1991 till now, the Department of Computer Science (DCS) offers Computer Science as a subject for the Physical Science students. In 2007, the DCS commenced a new degree programme named as Bachelor of Science Degree in Computer Science (BSc in Computer Science) for the students who are directly admitted by the UGC under the Computer Science (Direct-intake) stream. In 2015, an extended fourth year Applied Science degree in Computing was introduced for the Physical Science students who could not get into the special degree programme in Computer Science.

THE DEGREES OFFERED

The degree programmes at the DCS have been designed in accordance with the Sri Lanka Qualifications Framework (SLQF), which is a framework developed by UGC in order to strengthen the quality assurance mechanisms of the entire higher education sector in Sri Lanka. At present, the DCS prepares undergraduates for the award of degrees for the following degree programmes:

Stream	Name of the Degree	Duration	SLQF Level
Direct-intake	Bachelor of Science in Computer Science [BSc (Computer Science)]	3 years	5
	Bachelor of Science Honours in Computer Science [BSc Hons (Computer Science)]	4 years	6
	Bachelor of Science Honours in Applied Science in Computing [BSc Hons (AppliedSc)]	4 years	6
Physical Science	Bachelor of Science [BSc]	3 years	5
	Bachelor of Science Honours in Computer Science [BSc Hons (ComputerSc)]	4 years	6
	Bachelor of Science Honours in Applied Science in Computing [BSc Hons (AppliedSc)]	4 years	6

OBJECTIVES OF THE DEGREES

- Possess practical and theoretical knowledge of computer science and software engineering to contribute to the economic development of the region and nation.
- Prepared to achieve successful performance in postgraduate or professional degree programmes.
- Recognise the importance of and possess the skills necessary for life-long learning.

GRADUATE PROFILE

Students awarded BSc Hons (Computer Science) degree should be able to:

- Demonstrate mastery in core knowledge areas of Computer Science
- Analyse, design and develop sustainable solutions for real world problems while taking social, ethical and economic constraints into consideration.
- Express the ability to work effectively as an individual and a team member in software projects by meeting specified design and performance requirements
- Carry out scientific research by designing and conducting experiments, as well as by analysing and interpreting results
- Communicate competently and effectively with different levels of stakeholders
- Be lifelong learners of new trends in computing and focused on their career progress.

STRUCTURE OF THE DEGREE PROGRAMME

For the direct-intake Computer Science students, the DCS offers the following three degrees:

- BSc (Computer Science) degree of three-year duration
- BSc Hons (Computer Science) of four-year duration, and
- BSc Hons (AppliedSc) of four-year duration

The selection to BSc Hons (Computer Science) study programme is made at the end of second year, and for the BSc Hons in Applied Science degree programme is at the end of third year. Based on the preference of the students and the performance in their examinations, a limited number of students will be admitted to follow a four-year degree programme.

In Level-1S and Level-2S, students shall follow all the course units aggregating to *sixty credits*. The number of credits for Level-3S shall be 33 for those students who pursue BSc Honours in Computer Science degree programme. These students shall offer 27 credits in Level-4S as they have to carry out a four to six months industrial training during the second semester. When a student decides to opt with Level-3S or would like pursue the BSc Hons in Applied Science degree programme, the best 30 credits out of 33 of Level-3S

shall be considered when calculating the GPA of the 3S student. The students who pursue BSc in Computer Science degree programme have a freedom to omit one of the following advanced course units: CSC306S3 / CSC307S3 / CSC308S3 / CSC309S3 / CSC310S3 / CSC311S3 / CSC312S3.

If a student applied for all the Level-3S courses aggregating to 33 credits, then all the 33 credits will be taken into account when calculating the OGPA for the award of degree.

Note: Please refer to the Undergraduate Student Handbook issued by the Office of the Dean, Faculty of Science to get to know details about the Supplementary subjects, English literary course, Evaluation procedures and examinations, Examination offenses and punishments, Criteria for awarding degrees, Awarding of classes, and Effective date of the qualifications and official transcripts.

LIST OF COURSE UNITS OFFERED

Level	Unit Code	Unit Title	No. of Hours	
			Theory	Practical
1S	CSC101S3	Foundations of Computer Science	45	--
	CSC102S3	Computer Programming I	--	135
	CSC103S3	Introduction to Computer Systems	30	30
	CSC104S2	Mathematics for Computing I	30	--
	CSC105S3	Statistics for Computing I	45	--
	CSC106S3	Human Computer Interaction	30	30
	CSC107S2	Multimedia Technologies	30	--
	CSC108S2	Design of Algorithms	30	--
	CSC109S2	Introduction to Computer Security and Cryptography	30	--
	CSC110S2	Organisational Behaviour	30	--
	CSC111S2	Mathematics for Computing II	30	--
	CSC112S3	Statistics for Computing II	45	--
2S	CSC201S2	Database Systems Concepts and Design	30	--
	CSC202S2	Computer Programming II	--	90
	CSC203S2	Operating Systems	30	--
	CSC204S2	Data Structures & Algorithms	30	--
	CSC205S2	Software Engineering	30	--
	CSC206S4	Mathematics for Computing III	60	--
	CSC207S3	Computer Architecture	30	30

	CSC208S3	Concepts of Programming Languages	30	30
	CSC209S3	Bioinformatics	30	30
	CSC210S3	Web Technologies	30	30
	CSC211S2	Emerging Trends in Computer Science	15	30
	CSC212S2	Professional Practice	30	--
3S	CSC301S3	Rapid Application Development	45	--
	CSC302S2	Computer Programming III	--	90
	CSC303S2	Data Communication and Computer Networks	30	--
	CSC304S3	Team Software Project	20*	--
	CSC305S2	Graphics and Visual Computing	30	--
	CSC306S3	Advanced Database Design and Systems	45	--
	CSC307S3	Advanced Topics in Computer Networks	45	--
	CSC308S3	Artificial Intelligence	30	30
	CSC309S3	High Performance Computing	30	30
	CSC310S3	Image Processing and Computer Vision	30	30
	CSC311S3	Machine Learning	30	30
	CSC312S3	Mobile Computing	45	--

*Mentoring hours

Level	Unit Code	Unit Title	No. of Hours	
			Theory	Practical
4S	CSC401S3	Advanced Algorithms	45	--
	CSC402S3	Compiler Design	45	--
	CSC403S3	Data Science	30	30
	CSC404S3	Information Systems Security	45	--
	CSC405S3	Systems and Network Administration	15	60
	CSC406S6	Research Project	15	585†
	CSC407S6	Industrial Training	15	585†

†Notional hours

HUMAN RESOURCES OF THE DCS

ACADEMIC STAFF

Dr. S. Mahesan

Senior Lecturer, Gr. I
B.Sc. [Hons] (Jaffna), M.Sc. (Wales), Ph.D. (Cardiff)
MIEEE

Dr. E. Y. A. Charles

Senior Lecturer, Gr. I
B.Sc. [Hons] (Jaffna), Ph.D. (Cardiff)
MIEEE, MACM

Dr. K. Thabotharan

Senior Lecturer, Gr. I
B.Sc. [Hons] (Jaffna), M.Sc. (Akron), Ph.L. (Uppsala), Ph.D. (Colombo)
MIEEE, MACM

Dr. A. Ramanan

Senior Lecturer, Gr. I
B.Sc. [Hons] (Jaffna), Ph.D. (Southampton)
MIEEE

Dr. (Mrs). B. Mayurathan

Senior Lecturer, Gr. I
B.Sc. [Hons] (Jaffna), Ph.D. (Peradeniya)
MIEEE

Mr. S. Suthakar

Senior Lecturer, Gr. I
B.Sc. [Hons] (Jaffna), M.Phil. (Jaffna)
MIEEE

Dr. M. Siyamalan

Senior Lecturer, Gr. II
B.Sc. [Hons] (Jaffna), MSc (Nice), PhD (Dundee)
MIEEE

Mr. K. Sarveswaran

Lecturer
B.Sc. [Hons] (Peradeniya), M.Sc. (Moratuwa)
MIEEE

Dr. S. Shriparen

Senior Lecturer (Gr. II)
B.Sc. [Hons] (Jaffna), M.Sc. (Colombo)
MIEEE

ACADEMIC STAFF (CONT…)

Dr. T. Kokul

Senior Lecturer Gr.II
B.Sc. [Hons] (Jaffna), Ph.D. (QUT, Australia)
MIEEE

Dr. (Ms.) R. Nirthika

Lecturer (Probationary)
B.Sc. [Hons] (Jaffna)
MIEEE

Ms. J. Samantha Tharani

Lecturer (Probationary) [On Study Leave]
B.Sc. [Hons] (Jaffna)
MIEEE

Ms. M. Mayuravaani (On Study Leave)

Lecturer (Probationary)
BSc Hons (Computer Science) (Jaffna)
MIEEE

ACADEMIC SUPPORT STAFF

Mr. T. Sugirthan

Assistant Network Manager, Gr. I
B.Sc. (Jaffna), HNDIT,
M.Sc. (Peradeniya)

Mr. V. Visithan

Programmer Cum Systems Analyst, Gr. I
B.Sc. [Hons] (Jaffna),

NON-ACADEMIC STAFF

Mr. N. Thileepan

Management Assistant, Gr. I

Mrs. A. Vahini

Management Assistant, Gr. III
B.B.A. [Hons] (Jaffna)

Mr. Y. Hajanthan

Technical Officer, Gr. II
Seg."B"
B.Sc. (Bio Science)

Mr. A. Arulnesan

Laboratory Attendant, Lower Grade

INDUSTRIAL LINKAGES

The curriculum of the DCS incorporates an Industrial Training in their final year. Students are expected to be trained in Computing industries for a period of Four to Six months under the guidance of academic and industrial supervisors. Students shall maintain a journal to record his/her progress activity during the training. During the industrial training, students are expected to apply acquired knowledge in the industrial environment, develop interpersonal, communication, management and team working skills, adapt to work readily in real industrial projects, and perceive state-of-the-art industrial technologies. Four batches of students obtained industrial training since 2017 in the following industries:

Industry	Location
#native	Colombo
91 Solutions	Kattubetha
Aasa IT Solutions	Dehiwala
Acmi Group	Colombo
Addverb Technologies	India
Airport Aviation	Katunayake
Apptimus Tech	Jaffna
Arima Technologies	Jaffna
Arimac	Colombo
Axienta	Colombo
Ayapa Tech	Colombo
Black Vault Technologies	Colombo
Blockchain AI Pvt Ltd	Dehiwala
Burgeon Solutions	Pita Kotte
Ceymplon	Jaffna
Creative Software	Colombo
Cubo Systems	Kotte
Dewan Media	Malabe
Dialog Axiata	Colombo
Digiratina Technology	Dehiwala
Ellipsis	Nugegoda
EmetSoft	Colombo
hSenid Mobile Solutions	Colombo
hSenid Software International	Colombo
HuEx Studio	Colombo
IFS	Colombo
Innovay	Jaffna
Link Lanka	Colombo

Industry	Location
Liveroom	Nugegoda
Micronet Global Service	Colombo
Mitra	Moratuwa
MyDynamica	Jaffna
Neurotechnology Lab	Colombo
nCinga Innovations	Colombo
Oddly	Colombo
Onimta Information Technology	Maharagama
Peercore Software Solutions	Colombo
Prime Technologies	Nelliady
Pulz Technologies	Colombo
RedLabs	Pannipitiya
Redot Global	Rathmalana
SK International Holdings	Kotte
Softknowedge Business Solutions	Kurunegala
Speed IT Net	Jaffna
Sri Lanka Naval Headquarters	Colombo
Sri Lanka Telecom	Colombo
SUTD	Singapore
Sysco Labs	Colombo
Techorin	Colombo
Unicom SD	Jaffna
Virtusa	Colombo
VisuaMatix	Nawala
Wavenet	Colombo
Way Forward Energy	Nugegoda
WSO_2	Colombo
Yarl IT Hub	Jaffna

STUDENT LEARNING RESOURCES AND SUPPORT SERVICES

LEARNING RESOURCES

- The Main Library of the University (www.lib.jfn.ac.lk)
- Computer Unit of the University (www.cu.jfn.ac.lk)
- Department of English Language Teaching (DELT) (www.arts.jfn.ac.lk/index.php/home-eltc)
- Learning Management System (LMS) (http://lms.jfn.ac.lk/lms)

SUPPORT SERVICES

- Career Guidance Unit (CGU) (https://sites.google.com/univ.jfn.ac.lk/career-guidance-unit)
- Centre for Gender Equity and Equality (CGEE) (www.unit.jfn.ac.lk/cgee)
- Health Centre of the University (www.jfn.ac.lk/index.php/health-centre)
- Physical Education Unit (PEU) to facilitate Sports Activities (www.unit.jfn.ac.lk/peu)
- Students' Complex of the University consists of Bank, Canteens, CGU, Offices of the Senior Student Counsellor and Marshal, Post Office, Prayer Rooms, Stationery Shop, Students' Union Offices, Student Welfare Services Branch, UBL, and WeBe Centre.
- Student Hotspot (Open Study Hall with Wi-Fi Access near the Mathematics Block)
- Student Welfare Services Branch
- University Business Linkage (UBL) (https://sites.google.com/view/ubl-jaffna/home)
- Well-Being (WeBe) Centre (https://sites.google.com/univ.jfn.ac.lk/well-being-centre)

SUPPORT SYSTEMS AT DCS

- Useful links in the DCS website (www.csc.jfn.ac.lk)
- Computer Science Student Progress Management System (CS-SPMS)

  An up-to-date examination and assessment results are maintained in the CS-SPMS through which students can monitor their progress. (www.apps.jfn.ac.lk/csspms)
- Computer Science Student Attendance Management System (CS-SAMS)

  Student attendance to every course offered by the DCS is maintained in the CS-SAMS through which staff can monitor student progress. (www.apps.jfn.ac.lk/cssams)
- CompSoc Park to facilitate students' self/group learning and leisure activities
- Learning materials in LMS (http://lms.jfn.ac.lk/lms/course/index.php?categoryid=9)
- Open Educational Resources (OER) and other online resources (www.csc.jfn.ac.lk/onlineres)

RAGGING COMPLAINT PORTAL

The University Grants Commission (UGC) has set up an online portal for any member of the university community to lodge a complaint regarding incidents of ragging, harassment, intimidation and bullying. All complaints that are lodged will be investigated and victims of such incidents will be offered support and redress. The portal can be accessed via https://eugc.ac.lk/rag

PRIZES

PROFESSOR KANDIAH KUNARATNAM MEMORIAL PRIZE FOR COMPUTER SCIENCE

Founded in 2017 by the children of Professor Kunaratnam in memory of their father Professor Emeritus Kunaratnam. This prize is awarded annually to the student among the direct-intake students having the Best Performance obtaining highest OGPA in B.Sc. Hons Degree Examination in Computer Science held at the end of Level-4S and with a First Class or Second Class Upper Division. In case of tie, the GPA in Level-4S shall be considered. The best student is awarded with a certificate and an amount of cash reward.

UNIVERSITY PRIZE

- University Prize for having the best performance in Bachelor of Science Degree Examination in Computer Science in each Level-1S, Level-2S, and Level-3S.
- University Prize for having the best performance in Bachelor of Science Honours Degree Examination in Computer Science in Level-4S

DEAN'S LIST

The Dean's List is to recognize the level of high scholarship demonstrated by undergraduate students in each level of study (Level-1S, Level-2S, Level-3S, and Level-4S). The Dean's list is awarded at the end of each academic year to those students who possess a GPA of at least 3.70.

STUDENT ENTITIES

COMPUTER SOCIETY

The Computer Society of the University of Jaffna, in short known as CompSoc, is a dynamic organisation of enterprising individuals brought together by their common interest in Computing. Furthermore, students perform several activities through this society such as releasing a technical newsletter named as "Kananiyam" once a year, conducting seminars to recap ICT subjects and past examination papers to those students who prepare to sit for the G.C.E. (O/L) and G.C.E. (A/L) examinations, and organising cultural activities. All Computer Science students shall join as a member of the society. More details at http://society.jfn.ac.lk/comsoc.

IEEE STUDENT BRANCH OF THE UNIVERSITY OF JAFFNA

The Institute of Electrical and Electronics Engineers (IEEE) is the world's largest technical professional organisation dedicated to advancing technology for the benefit of humanity. The DCS has established the IEEE Student Branch of the University of Jaffna (IEEE Student Branch, UoJ) in September 2018 to provide and promote the theory as well as the practice of all aspects of Computer Science, Computer Engineering, Information and Communication Technology, Electrical, and Electronics. At present, there are 30 active student members consisting of four graduate student members, 21 student members from the DCS, and five student members from the Department of Electrical Engineering, Faculty of Engineering, University of Jaffna. More details at http://society.jfn.ac.lk/ieee.

GAVEL CLUB

The DCS recently established a Gavel Club to enhance communication and leadership skills among students through experiential learning. The Gavel Club attempts to empower students in the Faculty of Science to achieve their full potential and realise their dreams to transform into better individuals. Activities performed during monthly meetings provide a mutually supportive and positive learning environment in which every individual member has the opportunity to develop oral communication and leadership skills, which in turn foster self-confidence and personality development.

EVENTS AT DCS

SRI LANKA STUDENT WORKSHOP ON COMPUTER SCIENCE (SL-SWCS)

The first student-focused national workshop on Computer Science referred to as "Sri Lanka Student Workshop on Computer Science (SL-SWCS)" is a proud initiative by the DCS. SL-SWCS provides a stimulating opportunity for young research students across the country to get involved in discussions with researchers from local and foreign institutions. Students get an opportunity to present the results of their research activities via poster presentations in a friendly and constructive environment that would possibly lead to future collaborations with their scientific peers. The first national workshop SL-SWCS'11 was held on 8th of December 2011 in Jaffna. Since then SL-SWCS is conducted as a biennial workshop that is well attended by around 150 participants. More details at www.csc.jfn.ac.lk/index.php/slswcs

CS-CAREER FAIR

The DCS conducts a Career Fair that we refer to as "CS-Career Fair" in order to facilitate our Level-3 students in securing placements for industrial training and for Level-4 students to improve their employability. The first Career Fair was conducted on 5th of August 2017, the second event on 6th of October 2018, and the third event on 26th of October 2019 at the DCS. More details at www.csc.jfn.ac.lk/index.php/career-fair

UOJCODERS

The University of Jaffna (UoJ) coding competition that we refer to as "UoJCoders" is aimed to enhance the problem-solving skills and programming skills among the undergraduate students in computing to make them efficient programmers. The competition is a 12-hour inter-university coding competition

inspired by IEEEXtreme. The competition is open to all undergraduate students in Sri Lanka, especially for those who follow degrees in the computing discipline. UoJCoders v1.0 was held on 16th and 17th of March 2019 in the Department of Computer Science, University of Jaffna. The 12-hour coding competition was conducted online using the *hackerrank platform (http://hackerrank.com)*. More details at www.csc.jfn.ac.lk/index.php/uojcoders. UoJCoders v2.0 was planned to be held in April 2020 but due to the COVID-19 pandemic situation the event is postponed and will be conducted in October 2020.

INDUSTRIAL VISIT

The DCS has built strong linkages with Computing industries since 2010. The DCS will organise a one or two days' industrial visit to Computing industries for the Level-2S students with the support of the Sri Lanka Association of Software and Service Companies (SLASSCOM). From 2010 to year 2015, students of the DCS have visited Computing industries, namely, 99x Technology, CodeGen, hSenid Mobile Solutions, IFS, Millennium IT, MicroImage, Pearson Lanka, Virtusa, WSO_2, and Zone 24x7. This visit enables students to know about the infrastructure facilities that are in IT industries and to discuss the latest trends in software development.

TECH-TALKS/WEBINARS

Technical talks (Tech-Talks) or Webinars cover technical concepts and ideas that make it easier for students and staff to update their knowledge to the state-of-the-art techniques used in the areas of Computing. The DCS in collaboration with the CompSoc / IEEE Student Branch, UoJ organises a number of TechTalks/Webinars every year. More details can be found at:

www.csc.jfn.ac.lk → Activities → Tech-Talks/Webinars.

Level–1S

Effective from the Academic Year: 2016/2017

Department of Computer Science

Faculty of Science

University of Jaffna

Sri Lanka

Developed in February 2018

Course Code:	**CSC101S3**		
Course Title:	**Foundations of Computer Science**		
Credit Value:	**03**		
Core/Optional:	**Core**		
Hourly Breakdown:	Theory	Practical	Independent Learning
	45	--	**105**

Objectives:

Provide logical and mathematical foundations of computer science, and illustrate the use of formal languages in computer science

Intended Learning Outcomes:

- Describe the fundamentals of mathematical and logical aspects
- Outline the concepts of programming
- Illustrate the use of formal languages in computer science
- Explain basic computer network organisation

Course Contents:

- Fundamental organisation of computer hardware and software: Motherboard, I/O Peripherals, Expansion slots and cards, application software, arithmetic-logic unit, registers, central processing unit, memory, storage devices
- Theoretical foundations of sets: Basic notation, representations and examples, membership and subsets, operations on sets, Cartesian products, power sets, cardinality, infinite sets
- Introduction to relations and functions: Domain and range of a relation, one-to-one, one-to-many, many-to-one, inverse, reflexive, symmetric, transitive relations, into, onto, one-one, Bijective functions.
- Concepts of flowcharts and algorithms
- Introduction to propositional and predicate logic: Propositions, quantifiers, predicates, arguments
- Boolean algebra and logic gates: Combinatorial circuits, Boolean functions, Karnaugh map
- Number systems and their representations: Representation of integers and floating-point numbers in signed-magnitude and two's-complements
- Trees, Graphs and their applications: Graphs, representation of graphs, paths and circuits, planar graph, Binary trees, decision trees, tree traversal, spanning trees
- Automata, grammars and languages: Finite state machines, languages and grammars, language recognition, Turing machines
- Introduction to computer networks: Network topologies, transmission media and network devices, ISO OSI stack, IP addressing

Teaching/Learning Methods:

Use of chalkboard, Vocabulary drills, Flowcharts, Recitation oral questions, Timelines, Photographs, Tutorial discussions

Assessment Strategy:

- In-course Assessments 30%
- End-of-course Examination 70%

References:

- B. A. Forouzan, Foundations of Computer Science, 3rd Ed., 2014.
- P. Norton, Introduction to Computers, 7th Ed., Tata McGraw Hill Education, 2011.
- R. L. Graham, Donald E. Knuth, and Oren Patashnik. Concrete Mathematics: Foundation for Computer Science, 2nd Ed., Addison-Wesley Professional, 1994.

Course Code:	**CSC102S3**		
Course Title:	**Computer Programming I**		
Credit Value:	**03**		
Core/Optional:	**Core**		
Hourly Breakdown:	Theory	Practical	Independent Learning
	--	**135**	**165**

Objectives:

Provide fundamentals of programming concepts in Java and introduce the concepts of object-oriented programming

Intended Learning Outcomes:

- Demonstrate fundamental programming concepts
- Identify classes, objects, members of a class and relationships among them needed for a specific problem
- Solve variety of computational problems
- Create programs using fundamental concepts of object-oriented programming

Course Contents:

- Interaction with a computer: Graphical user interface, command line interface, files and folders, using text editors / IDEs, programming principles
- Program development in Java: Programming principles, Edit-Compile-Run cycle, basic components of a Java program, syntax and semantics, data types, variables and constants, expressions, built-in classes
- Introduction to Object-Oriented Programming: Classes and objects, fields and methods, arguments and parameters, constructors, class and instance data values
- Control Flow: Sequence, selection, repetition, explicit control-flow statements
- Arrays and Collections: 1D & 2D Arrays, arrays of objects, for-each loop, passing arrays to methods, searching and sorting in arrays
- Concepts of recursion and backtracking: Recursion concepts, examples using recursion, recursion vs. iteration, recursive backtracking
- Implementing standard algorithms: String matching, counting coins, Knapsack problem, Huffman coding, activity-selection, scheduling problem
- Inheritance and Polymorphism: Principles of Inheritance, member accessibility, inheritance over accessibility, principles of polymorphism and polymorphic constructors/methods

Teaching/Learning Methods:

Laboratory experiments, Supervised study, Practical records, Tutorial discussions

Assessment Strategy:

- In-course Assessments
  - Assessment on practical records 10%
  - End-of-First Semester Practical Assessment 30%
- End-of-Second Semester Practical examination 60%

References:

- C. T. Wu, An Introduction to Object-Oriented Programming with Java, 5th Ed., McGraw-Hill Education, 2009.
- P. Deitel and H. Deitel, Java How to Program, 9th Ed., Pearson Education, Inc., 2012.
- H. Schildt, Java: The Complete Reference, 9th Ed., McGraw-Hill Osborne Media, 2014.

Course Code:	**CSC103S3**		
Course Title:	**Introduction to Computer Systems**		
Credit Value:	**03**		
Core/Optional:	**Core**		
Hourly Breakdown:	Theory	Practical	Independent Learning
	30	**30**	**90**

Objectives:

Provide the students a conceptual level understanding of the structure and operation of computers and the Internet. In addition, this course also provides a basic understanding on threats to computer systems and computer networks

Intended Learning Outcomes:

- State the conceptual and physical structure of a computer
- Describe the operation of a computer system and the components
- Demonstrate the structure and operations of the Internet
- Troubleshoot a computer for hardware and software related issues

Course Contents:

- Conceptual design and operation of modern computers: Data and Information, Conceptual design of modern computers (John von Neumann) and stored program concept, CPU and memory organisation, execution of programmes, Booting process, digital storage devices
- Basics of the Internet: Structure of the Internet, Operation of Internet – TCP IP, IP Addressing, Domain naming system, Role of servers and clients, Management and control of the Internet – ISP, Internet Consortium
- Internet services and applications: WWW, e-mail, e-learning, Social Networking, Blogs, Cloud computing
- Threats to computer systems and information: Computer malware, ways to protect computer systems from malware, securing information - encryption technique, digital signature, biometric devices, email filtering, firewall, and precautions on Web; prevention of electronic theft
- Computer related ethical issues: copyright, software licenses, information privacy, intellectual property, content filtering, Spam, and laws enacted with regards to SPAM, protecting web and electronic communication from antisocial and anti-cultural elements
- Maintaining Computer hardware and software: Servicing a computer, installing operating systems and configuring a computer, installing software and drivers, troubleshooting computers, setting up a computer network

Teaching/Learning Methods:

Lectures, Workshop sessions, Visit to Network Operating centre, Tutorial discussions

Assessment Strategy:

- In-course Assessment (Theory) 15%
- In-course Assessment (Practical) 15%
- End-of-course Examination 70%

References:

- P. Norton, Introduction to Computers, 7th Ed., Tata McGraw Hill Education Private Limited, 2011.
- R. White and T.E. Downs, "How Computers Work", 3rd Ed., 2008.

Course Code:	**CSC104S2**		
Course Title:	**Mathematics for Computing I**		
Credit Value:	**02**		
Core/Optional:	**Core**		
Hourly Breakdown:	Theory	Practical	Independent Learning
	30	--	**70**

Objectives:

Provide fundamental mathematical concepts and techniques in finding the solution methods for real world problems and then to apply those techniques for the proposed solutions by a computer

Intended Learning Outcomes:

- Apply proof techniques in the construction of a sound argument
- Perform the operations associated with sets
- Analyse connections between various types of relations
- Classify various types of functions
- Illustrate the use of Boolean algebra in logic circuit designs

Course Contents:

- Proof Techniques: Notions of implication, equivalence, converse, inverse, contrapositive, negation, and contradiction. Direct proofs, Disproving by counterexample, Proof by contradiction
- Set theory: Venn diagrams, set operations, Cartesian product, Power sets, Cardinality of finite sets
- Relations and functions: Reflexivity, symmetry, transitivity of relations, Equivalence relations, partial orders; Surjections, injections, bijections; Inverses, Composition of functions
- Boolean algebra: Introduction, Duality, Representation theorem, Sum-of-products from Boolean algebra

Teaching/Learning Methods:

Lectures, class discussions, textbook assignments

Assessment Strategy:

- In-course Assessments 30%
- End-of-course Examination 70%

References:

- S. Lipschutz, "Set theory and related topics", McGraw-Hill, 1998.
- R. R. Stoll, "Set theory and logic", 1979.

Course Code:	**CSC105S3**		
Course Title:	**Statistics for Computing I**		
Credit Value:	**03**		
Core/Optional:	**Core**		
Hourly Breakdown:	Theory	Practical	Independent Learning
	45	--	**105**

Objectives:

Provide a solid theoretical foundation of Statistics with a combination of experience in solving real world problems with a computer

Intended Learning Outcomes:

- Describe the concepts of probability
- Apply Bayes' Theorem
- Utilise the probability distributions for real world problems
- Describe the concept of Random variable

Course Contents:

- Introduction to probability: Permutations, combinations, Venn diagram, events, sample space, mutually exclusive events, axioms of probability, laws of probability, conditional probability, independence
- Bayes' Theorem and Applications: Partition, total probability theorem, Bayes' theorem, tree diagram
- Random variable: Discrete and continuous random variables, probability mass function, probability density function, expectation, moments, mean and variance, moment generating functions, probability generating functions
- Probability distribution: Discrete uniform, Bernoulli, binomial, Poisson, geometric, uniform, exponential and normal distributions, applications of the normal distribution, sampling distribution of the sample means
- Joint distributions: Joint distributions, marginal distribution, conditional distributions, conditional expectation and variance

Teaching/Learning Methods:

Lectures, class discussions, textbook assignments, Guided learning

Assessment Strategy:

- In-course Assessments 30%
- End-of-course Examination 70%

References:

- S. Ghahramani, "Fundamentals of Probability", 2004.
- M. R. Spiegel, "Probability Schaum's Outline Series", 2000.

Course Code:	**CSC106S3**		
Course Title:	**Human Computer Interaction**		
Credit Value:	**03**		
Core/Optional:	**Core**		
Hourly Breakdown:	Theory	Practical	Independent Learning
	30	**30**	**90**

Objectives:

Introduce principles and methods to build effective computer interfaces for users

Intended Learning Outcomes:

- Explain the fundamental theories of Human Computer Interaction (HCI), key features of interaction and common interaction styles.
- Evaluate heuristic user interfaces by using a variety of analysis and design methods
- Apply user-centered and contextual design techniques for human computer interface design scenarios
- Implement a prototype of a user interface for a system that satisfies usability requirements
- Discuss the design of HCI in various recent developments

Course Contents:

- **Foundations of Human Computer Interaction (HCI) and the Design Process:** Human Capabilities, Interaction Design Basics, HCI in the Software Process, Design Rules, Universal Design, The Human Body and Device Design
- **Human Cognition and Interaction Styles:** Goals, Operators, Methods, and Selection (GOMS), Keystroke-Level Modelling, Time-scales and the Illusion of Multitasking, Metaphor, Direct Manipulation, Command Languages
- **Usability Engineering:** Observing Users, Usability Analysis - Error Handling, Error Prevention, Cognitive Walkthroughs, Heuristic Evaluation, Usability Guidelines, Usability Methods; Prototyping, Task Analysis, User-Centred Design
- **User Interface Programming:** Interface Implementation, Events and Handlers, Development Tools
- **Recent Development in HCI:** Groupware, Ubiquitous Computing, Virtual and Augmented Realities, Hypertext and Multimedia

Teaching/Learning Methods:

Lectures, Modelling in various media, Construction of diagrams, Case studies

Assessment Strategy:

- In-course Assessments (Theory) 15%
- In-course Assessments (Practical) 15%
- End-of-course Examination 70%

References:

- H. Sharp, Y. Rogers, and J. Preece. Interaction Design: Beyond human-computer interaction, 4th Ed., Wiley Publishers, 2015.
- B. Shneiderman, C. Plaisant, M. Cohen and S. Jacobs. Designing the User Interface: Strategies for Effective Human-Computer Interaction, 5th Ed., Addison Wesley publishers, 2010.

Course Code:	**CSC107S2**		
Course Title:	**Multimedia Technologies**		
Credit Value:	**02**		
Core/Optional:	**Core**		
Hourly Breakdown:	Theory	Practical	Independent Learning
	30	--	**70**

Objectives:

Provide in-depth knowledge in technologies to develop multimedia-based contents

Intended Learning Outcomes:

- Explain the fundamental principles of multimedia
- Demonstrate compression techniques used in multimedia
- Discuss theories behind the multimedia components
- Design contents using multimedia technologies

Course Contents:

- Introduction: Uses of multimedia, interaction technologies, multimedia hardware and devices
- Compression techniques in multimedia: compression basics, lossless and lossy compression techniques
- Text in multimedia: Visual representation of text, digital representation of characters
- Fundamentals of colours, colour models and dithering
- Fundamentals of images: characteristics of images, image file formats, and image compression standards
- Digital audio: sound processing, representation of audio files
- Fundamentals of video and animation: analogue and digital video standards, video processing, video compression standards and file formats, basics of animation
- Designing multimedia contents: Development phases, multimedia authoring and tools, multimedia in the internet

Teaching/Learning Methods:

Lectures, recitation of oral questions, use of chalkboard and multimedia presentations

Assessment Strategy:

- In-course Assessments 30%
- End-of-course Examination 70%

References:

- Z.N. Li and M.S. Drew, "Fundamentals of Multimedia", 2nd Ed., 2014.
- A. Banerji; A. M. Ghosh, "Multimedia technologies", 2010.
- T.M. Savage and K.E. Vogel, "An Introduction to Digital Multimedia", 2nd Ed., 2013.

Course Code:	**CSC108S2**		
Course Title:	**Design of Algorithms**		
Credit Value:	**02**		
Core/Optional:	**Core**		
Hourly Breakdown:	Theory	Practical	Independent Learning
	30	--	**70**

Objectives:

Provide in-depth knowledge in different algorithmic approaches for problem solving

Intended Learning Outcomes:

- Demonstrate familiarity of various algorithm design techniques and their applications
- Use different strategies to compare the performance of algorithms
- Discuss the usages of different Iterative and recursive algorithms
- Apply different algorithmic approaches and concepts for solving computational problems

Course Contents:

- Algorithm Analysis: Informal comparison of algorithm efficiency, best, expected, and worst case behaviours, time and space trade-offs in algorithms, Asymptotic analysis (big O, little o, big Ω and big Θ notations)
- Problem-solving strategies: Iterative and recursive algorithms
- Brute-force and Greedy methods: Concepts of Brute-force and Greedy methods, applications of Brute-force and Greedy methods for solving problems (String matching, counting coins, Knapsack problem, Huffman Coding, Activity-selection, Scheduling problems)
- Searching and sorting in arrays and their complexities: Linear search, binary search, selection sort, insertion sort, bubble sort

Teaching/Learning Methods:

Lectures, class discussions, textbook assignments, laboratory practical

Assessment Strategy:

- In-course Assessments 30%
- End-of-course Examination 70%

References:

- T. Cormen, C. Leiserson, R. Rivest, C. Stein, Introduction to Algorithms, 3rd Ed., MIT Press, 2009.
- R. Sedgewick and K. Wayne, Algorithms, 4th Ed., Addison Wesley Publishers, 2011.

Course Code:	**CSC109S2**		
Course Title:	**Introduction to Computer Security and Cryptography**		
Credit Value:	**02**		
Core/Optional:	**Core**		
Hourly Breakdown:	Theory	Practical	Independent Learning
	30	--	**70**

Objectives:

Introduce Computer security principles and cryptography, and analyse the need of Public key infrastructure and its applications

Intended Learning Outcomes:

- Describe the concepts of computer security and cryptography
- Explain notions of threat, vulnerability, and attacks
- Utilise different symmetric key and public key encryption algorithms
- Discuss the importance of Public key Infrastructure and its applications

Course Contents:

- Foundations of computer security: introduction to computer security and cryptography
- Concepts of risks, threats, vulnerabilities and types of attacks: computer security attacks, types, attacker goals, motivations (such as underground economy, digital espionage, cyberwarfare, insider threats, hacktivism, advanced persistent threats), examples of malware, organizational vulnerabilities and threats
- Key security properties: confidentiality, integrity and availability
- Concepts of authentication, authorization and access control: OSI security architecture, security services and security mechanisms
- Classical encryption techniques: Symmetric key ciphers, substitution techniques, transposition techniques, other classical encryption tools, key management techniques
- Public-key cryptography and its applications: introduction to PKI, hash functions, digital signatures and digital certificates, secure web browsing, e-commerce, secure authentication

Teaching/Learning Methods:

Lectures, Class discussion, Recitation oral questions, Lecture demonstration, Small group discussions,
Use of slides, Textbook assignments.

Assessment Strategy:

- In-course Assessments 30%
- End-of-course Examination 70%

References:

- W. Stallings, Cryptography and Network Security: Principles and Practice, 6th Ed., 2013.
- J. Katz and Y. Lindell. Introduction to Modern Cryptography, 2nd Ed., 2014.

Course Code	**CSC110S2**		
Course Title	**Organisational Behaviour**		
Credit Value:	**02**		
Core/Optional:	**Core**		
Hourly Breakdown:	Theory	Practical	Independent Learning
	30	--	**70**

Objectives:

Provide students with an awareness of the concept of organisational behaviour and determinants of human behaviour in organisation

Intended Learning Outcomes:

- Identify individual level, group level and organizational level factors which influence human behaviour at work
- Describe the ways of developing personality, changing attitudes and motivating employees
- Define leadership theories, communication skills and change management approaches
- Outline the stress and conflict management techniques

Course Contents:

Introduction to organisational behaviour, personality, values, attitudes, perception, learning and reinforcement, motivation in the workplace setting, group and interpersonal process, foundations of leadership, conflict and negotiation, essentials of interpersonal communication, organisational change management

Teaching/Learning Methods:

Lectures, guest lectures, class discussions, small group discussions, case studies

Assessment Strategy:

- In-course Assessments 30%
- End-of-course Examination 70%

References:

- F. Luthans, Organisational Behaviour, 12th Ed., 2010.
- J.W. Newstrom and K. Davis, Organisational Behaviour: Human Behaviour at Work, 2004.
- L.J. Mullins, Organisational Behaviour, 11th Ed., 2013.

Course Code	**CSC111S2**		
Course Title	**Mathematics for Computing II**		
Credit Value:	**02**		
Core/Optional:	**Core**		
Hourly Breakdown:	Theory	Practical	Independent Learning
	30	--	**70**

Objectives:

Provide a solid foundation of Mathematics to apply them to solve problems in Computer Science

Intended Learning Outcomes:

- Describe basic properties of integers
- Use Euclid's algorithm
- Solve systems of linear congruences
- Describe basics of finite group theory

Course Contents:

- Number theory: Introduction, integers, factors and Euclid's algorithm, unique factorization, linear congruences, Inverses, Chinese remainder theorem, Fermat's Theorem.
- Group theory: Definitions and examples, order of elements, subgroups, cosets and Lagrange's theorem, cyclic groups.

Teaching/Learning Methods:

Lectures, class discussions, textbook assignments

Assessment Strategy:

- In-course Assessments 30%
- End-of-course Examination 70%

References:

- D. Burton, "Elementary Number Theory", 2010.
- J.B. Fraleigh, "A First course in abstract algebra", 2002.
- G.A. Jones and J.M. Jones, "Elementary number theory", Springer, 1998.
- B. Baumslag and B. Chandler, "Group Theory", 1968.

Course Code	**CSC112S3**		
Course Title	**Statistics for Computing II**		
Credit Value:	**03**		
Core/Optional:	**Core**		
Hourly Breakdown:	Theory	Practical	Independent Learning
	45	--	**105**

Objectives:

Train students in applying statistical methods in proposing solutions for real world problems to be solved with computer.

Intended Learning Outcomes:

- Examine data using summary statistics and graphical methods
- Outline different methods of parameter estimation in Statistics
- Discuss the principles of hypothesis testing with applications
- Apply simple linear regression technique to real world issues

Course Contents:

- Descriptive Statistics: Types of data, population, sample, parameter, statistic, tabular and pictorial presentation of data, summary statistics, measures of central tendency and dispersion, skewness, kurtosis
- Point and Interval Estimation: Sampling distributions, central limit theorem, confidence intervals for one-sample, two–sample population characteristics, sample size calculation for parameter estimation, interpretation of confidence intervals
- Testing Hypotheses: Steps in hypothesis testing, level of significance, Type–I and Type–II errors,
- p–value, power of test, Z–test, t –test, χ^2 test, and F-test, goodness of fit test
- Simple linear regression: Correlation, simple linear regression, least square estimation, interpretation of regression parameters, application of simple linear regression

Teaching/Learning Methods:

Lectures, Tutorial discussions, Guided Learning

Assessment Strategy:

- In-course Assessments 30%
- End-of-course Examination 70%

References:

- D.S. Moore, G.P. McCabe and B. Craig, Introduction to the Practice of Statistics, 6th Edition, 2009.
- R.E. Walpole, R.H. Myers, S.L. Myers, K.E. Ye, "Probability and Statistics for Engineers and Scientists", 9th Edition, 2010.
- S.A. Lesik, "Applied Statistical Inference with Minitab", 2009.

Level–2S

Effective from the Academic Year: 2017/2018

Department of Computer Science
Faculty of Science
University of Jaffna
Sri Lanka

Developed in November 2018

Course Code:	**CSC201S2**		
Course Title:	**Database Systems Concepts and Design**		
Credit Value:	**02**		
Core/Optional:	**Core**		
Hourly Breakdown:	**Theory**	**Practical**	**Independent Learning**
	30	--	**70**

Objectives:

Introduce database system concepts and fundamentals necessary for designing, implementing, and manipulating databases.

Intended Learning Outcomes:

- State key characteristics of a database
- Develop conceptual models for databases
- Create efficient databases
- Apply query language to create and manipulate databases

Course Contents:

- Introduction to database concepts and architecture: File systems, database system concepts, three-schema architecture, classifications of database systems and database users
- Data Modeling: Entity Relationship model, relational model, network model, hierarchical model, object relational model, UML class diagrams.
- Relational database design: Relational model concepts, defining a relational schema from an ER diagram, basics of functional dependencies and normalization (1NF, 2NF, 3NF and BCNF)
- Developing and manipulating databases: Data development and manipulation using SQL, MySQL, PostgreSQL and MongoDB
- Relational algebra and relational calculus: Binary operations, Cartesian product, extended relational operator, tuple relational calculus and domain relational calculus
- File organization for conventional DBMS: Storage devices and their characteristics, file organization, fixed-length records, variable-length records, sequential file organization, indexed sequential access method
- Introduction to transaction management, concurrency control and recovery: Concept of transactions, concurrency in transaction processing, recovering databases from failure

Teaching/Learning Methods:

Lectures, Recitation oral questions, Guided learning, Tutorial discussions

Assessment Strategy:

- In-course Assessments 30%
- End-of-course Examination 70%

References:

- Ramez Elmasri and Shamkant B. Navathe, Fundamentals of Database Systems, 7th Ed, Addison-Wesley, 2015.
- C.J. Date, An Introduction to Database Systems, 8th Ed, Addison-Wesley, 2003
- Ramakrishnan and Gehrke, Database Management Systems, 3rd Ed., McGraw-Hill, 2003.

Course Code:	**CSC202S2**		
Course Title:	**Computer Programming II**		
Credit Value:	**02**		
Core/Optional:	**Core**		
Hourly Breakdown:	**Theory**	**Practical**	**Independent Learning**
	--	**90**	**110**

Objectives:

Develop proficiency in writing programs to solve computational problems using suitable data structures.

Intended Learning Outcomes:

- Implement appropriate data structures to manipulate data for various computational problems
- Devise programs to solve complex computational problems
- Create databases using database management systems
- Develop web based applications that interact with databases

Course Contents:

- Designing and Implementing algorithms: Recursion, backtracking, Divide-and-conquer, and Dynamic programming.
- Fundamental data structures and their applications: Arrays, Lists, Stacks, Queues, Linked lists, Trees, and Graphs
- Database design, modeling and development: SQL (MySQL, MariaDB) and NoSQL (MongoDB, PostgreSQL)
- Develop web based applications: Web development using HTML, CSS and Scripting languages (PHP, JavaScript, JQuery, NodeJS)

Teaching/Learning Methods:

Lectures, Laboratory practical sessions, Guided learning, Assignments, Continuous practical recordings

Assessment Strategy:

- In-course Assessments
  - Assessment on practical records 10%
  - End-of-First Semester Practical Assessment 30%
- End-of-Second Semester Practical examination 60%

References:

- P. Deitel and H. Deitel, Java How to Program (Early Objects), 10th Ed, Prentice Hall, 2014.
- R. Sedgewick and K. Wayne, Algorithms, 4th Ed., Addison Wesley Publishers, 2011.
- N. Karumanchi, Data Structures and Algorithms Made Easy: Data Structures and Algorithmic Puzzles, 5th Ed, 2016
- D. Kalemis, The Fundamental Concepts of Object-Oriented Programming, 2013.
- R. Elmasri and S. B. Navathe, Fundamentals of Database Systems, 7th Ed, Addison-Wesley, 2015.
- D. Bartholomew, Getting Started with MariaDB, 2nd Ed, 2015.

Course Code:	CSC203S2		
Course Title:	Operating Systems		
Credit Value:	02		
Core/Optional:	Core		
Hourly Breakdown:	Theory	Practical	Independent Learning
	30	--	70

Objectives:

Provide fundamental concepts and functionalities of operating systems.

Intended Learning Outcomes:

- Describe the objective and functions of modern operating systems
- Explain how resources (such as CPU, memory, storage, file and devices) are managed by the operating system
- Demonstrate the operations of a prototypical process manager
- Compare various techniques used for concurrency control

Course Contents:

- Introduction to operating system: Architecture of modern operating systems (OS), evolution of OS, OS operations and functionalities, and open source OS
- Processes and Threads: Concept of process, process states, process control block, schedulers, context switch, interprocess communication, process scheduling, overview of threads, multicore programming and multithreading models
- Concurrency: Process synchronisation (race condition, critical-section problem, mutex locks, semaphores, classic problems of synchronization and monitors), deadlock (characterization, prevention, avoidance, detection and recovery)
- Memory management: Swapping, memory allocation, fragmentation, paging, segmentation, virtual memory and address translation
- Storage management: Mass Storage, host attached storage, network attached storage, storage area network and RAID
- File and I/O Device management: File organization and access, file system security, device drivers, direct memory access and interrupt handling

Teaching/Learning Methods:

Lectures, Case studies, Use of chalkboard, Simulation, Recitation oral questions, Guided learning, Tutorial discussions

Assessment Strategy:

- In-course Assessments 30%
- End-of-course Examination 70%

References:

- W. Stalling, Operating systems: Internals and Design Principles, 8th Ed, Pearson, 2014.
- A. Silberschatz, P. B. Galvin, G. Gagne, Operating System Concepts, 9th Ed, 2013.
- S. Tanenbaum and H. Bos, Modern Operating Systems, 4th Ed, 2014.

Course Code:	**CSC204S2**		
Course Title:	**Data Structures and Algorithms**		
Credit Value:	**02**		
Core/Optional:	**Core**		
Hourly Breakdown:	**Theory**	**Practical**	**Independent Learning**
	30	--	70

Objectives:

Introduce common data structures and standard algorithms for solving various types of problems.

Intended Learning Outcomes:

- Analyse the correctness and the performance of complex algorithms
- Discuss the implementation of standard data structures
- Demonstrate skills in solving complex computational problems using suitable data structures
- Explain the divide-and-conquer paradigm and dynamic programming strategies and their usages

Course Contents:

- Proof of correctness of algorithms: Contrapositive and contradiction, Induction, and Loop invariants
- Recurrence relations: Analysis of iterative and recursive algorithms (Quick sort and merge sort, etc.)
- Fundamental Data Structures: Arrays, Lists, Stacks, Queues, Linked lists, Trees, and Graphs
- Algorithm Design and Implementation Techniques: Divide-and-conquer paradigm, Dynamic programming algorithms, Recursive, and backtracking
- Applications of Trees and Graphs: Binary search, Dijkstra's shortest path, minimum spanning tree

Teaching/Learning Methods:

Lecture, Class discussions, Tutorial discussions, Assignments

Assessment Strategy:

- In-course Assessments 30%
- End-of-course Examination 70%

References:

- T. Cormen, C. Leiserson, R. Rivest, and C. Stein, Introduction to Algorithms, 3rd Ed., MIT Press, 2009.
- R. Sedgewick and K. Wayne, Algorithms, 4th Ed., Addison Wesley Publishers, 2011.
- N. Karumanchi, Data Structures and Algorithms Made Easy: Data Structures and Algorithmic Puzzles, 5th Ed. 2016.
- S. S. Skiena, The Algorithm Design Manual, 2nd Ed., Springer, 2011.

Course Code:	CSC205S2		
Course Title:	Software Engineering		
Credit Value:	02		
Core/Optional:	Core		
Hourly Breakdown:	Theory	Practical	Independent Learning
	30	--	70

Objectives:

Introduce all phases of the life cycle of a software system including requirements analysis and specification, design, construction, testing, deployment, operation, and maintenance.

Intended Learning Outcomes:

- Discuss the software engineering principles and life-cycle
- Identify different roles played by personnel in software development and their responsibilities
- Construct software designs based on user requirements
- Apply appropriate techniques in software development, testing, maintenance, and evolution

Course Contents:

- Introduction to Software Engineering: Software characteristics, impact of software, importance of engineering approaches, challenges and ethics in software development
- Introduction to systems analysis and design: Types of systems (transaction processing, management information, decision support, etc.), need for systems analysis and design, the system development life cycle, roles played by different personnel in system development life cycle including the role of the systems analyst
- Software development process models: Waterfall model, prototyping model, spiral model, evolutionary model, iterative model and agile methodology
- Software requirements and specifications: Types of requirements, requirement gathering processes and techniques, documenting requirements
- Software analysis techniques: Data flow diagrams, data dictionaries, process specifications and structured decisions
- Software design techniques: Object-oriented design using UML, Agile methodologies using SCRUM
- Software testing: Development testing, test-driven development, release testing and user testing
- Software maintenance and evolution: Evolution processes, program evolution dynamics, software maintenance and legacy system management

Teaching/Learning Methods:

Lecture, case studies, Recitation oral questions, Small groups discussions, Guided learning

Assessment Strategy:

- In-course Assessments 30%
- End-of-course Examination 70%

References:

- I. Sommerville, Software Engineering, 10th Ed, 2015.
- K. E. Kendall and J. E. Kendall, System Analysis and Design, 9th Ed, 2013.
- R. E. Beasley, Software Engineering: Principles and Practices, 2nd Ed, 2015.

Course Code:	**CSC206S4**		
Course Title:	**Mathematics for Computing III**		
Credit Value:	**04**		
Core/Optional:	**Core**		
Hourly Breakdown:	**Theory**	**Practical**	**Independent Learning**
	60	--	140

Objectives:

Encourage computer science students more aware of the importance of linear algebra in various computer science topics.

Intended Learning Outcomes:

- Develop an understanding of the theory of vector spaces.
- Use the theory of linear transformations and their matrix representation
- Solve systems of linear equations and understand the conditions for the existence of solution
- Use determinations and spectral properties.

Course Contents:

Vectors in R^n norms and inner products in R^n Cauchy-Schwartz and triangular inequalities, Gram-Schmidt process. Elementary operations and elementary matrices, echelon and row reduced echelon matrices. Vector spaces, linear dependence, and independence, subspaces, basis and dimension, Steinitz replacement theorem. Linear transformations, matrix representation and change of base, column rank, row rank and nullity of matrix. Determinants and their properties, invertibility of a square matrix, Eigen values and Eigen vectors, characteristic polynomials, Cayley-Hamilton theorem, orthogonal, symmetric and skew symmetric matrices, quadratic forms, diagonalization, System of linear equations.

Teaching/Learning Methods:

Use of chalkboard, Tutorial, Textbook assignments, Guided learning

Assessment Strategy:

- In-course Assessments 30%
- End-of-course Examination 70%

References:

- Devi Prasad, Elementary Linear Algebra, 2nd Ed., Narosa Publishing House, New Delhi, 2012
- David Lay C, Linear Algebra and Its Applications, 4th Ed., Pearson (Addison Wesley) Publication, 2012;
- Seymour Lipschutz, Schaum's Theory and problems of linear algebra, 2011
- Datta K.B, Matrix and Linear Algebra, Prentice hall of India Pvt. Ltd, New Delhi–110001, 2003

Course Code:	**CSC207S3**		
Course Title:	**Computer Architecture**		
Credit Value:	**03**		
Core/Optional:	**Core**		
Hourly Breakdown:	**Theory**	**Practical**	**Independent Learning**
	30	45	75

Objectives:

Understand the design of a digital computer including the structure of a microprocessor, memory organisation and program execution cycle.

Intended Learning Outcomes:

- Explain the conceptual design and the organisation of a computer system
- Describe processor unit design and its operations
- Summarise memory and Input/output organisation
- Build Assembly language programs

Course Contents:

- Introduction to modern computer architecture: Architectural and technological design and development, and performance measures of a processor
- Instruction set architecture models: Instruction set architectures and design, memory locations and operations, addressing modes, instruction types, microprogramming
- Processing unit design: CPU basics, register set, data path, CPU instruction cycle, control unit design, instruction pipelining techniques
- Memory hierarchies and Input/output organisation: Memory structure and hierarchy, cache memory mapping, direct memory access, virtual memory, interrupt-driven I/O, and Input-Output interfaces
- Assembly language programming: Instructions mnemonics and syntax, assembler directives and commands, assembly and execution of programs

Teaching/Learning Methods:

Lecture, Programming practical sessions, Tutorial discussions, Assignments, Guided learning

Assessment Strategy:

- In-course Assessment (Theory) 10%
- In-course Assessment (Practical) 30%
- End-of-course Examination 60%

References:

- D. A. Patterson and J. L. Hennessy, Computer Organization and Design: The Hardware and Software Interface, Morgan Kaufmann Publishers, 5th Ed, 2013.
- M. Abo-El-Barr and H. El-Rewini, Fundamentals of Computer Organization and Architecture, A John Wiley & Sons Publication, 2004.
- W. Stallings, Computer Organization and Architecture, Prentice Hall Publishers, 10th Ed, 2015.

Course Code:	**CSC208S3**		
Course Title:	**Concepts of Programming Languages**		
Credit Value:	**03**		
Core/Optional:	**Core**		
Hourly Breakdown:	**Theory**	**Practical**	**Independent Learning**
	30	30	90

Objectives:

Provide an overview of the basic concepts that appear in modern programming languages, the principles that underlie the design of programming languages, and their features.

Intended Learning Outcomes:

- Describe the fundamental issues in the design and the use of major programming languages
- Demonstrate the differences of programming paradigms in different programming languages
- Discuss the use of formal methods for program verification
- Build concurrent and functional programs

Course Contents:

- Introduction: Programming domains, evaluation criteria for programming languages, influences on language design, programming language categories
- Introduction to syntactic and semantic description of programming languages
- Programming paradigms in different programming languages: Data types, Abstract data types, Data objects, Control structures, subprograms, lifetime and scope of variables and functions, object-oriented programming, exception handling
- Concurrency: Basics of concurrency, subprogram-level concurrency, monitors, message passing, threads
- Functional programming: Fundamentals and programming with functional programming languages

Teaching/Learning Methods:

Lectures, practical sessions, Tutorial discussions, Assignments

Assessment Strategy:

- In-course Assessment (Theory) 10%
- In-course Assessment (Practical) 30%
- End-of-course Examination 60%

References:

- R. W. Sebesta, Concepts of Programming Languages, Pearson, 2016.
- J. C. Mitchell, Concepts in Programming Languages, Cambridge University Press, 2003.
- C. Ghezzi and M. Jazayeri, Programming language concepts, 3rd Ed, 1997.

Course Code:	**CSC209S3**		
Course Title:	**Bioinformatics**		
Credit Value:	**03**		
Core/Optional:	**Core**		
Hourly Breakdown:	**Theory**	**Practical**	**Independent Learning**
	30	30	90

Objectives:

Provide theoretical and practical knowledge in Bioinformatics including analysis of protein and genome sequences by various computational tools.

Intended Learning Outcomes:

- Describe computational genomics and phylogenetics concepts
- Demonstrate the use of computational tools for sequence analysis in bioinformatics
- Perform Data analysis and Pattern recognition in biological data
- Formulate a biological problem as a computational problem

Course Contents:

- Introduction to bioinformatics: Aims and tasks of bioinformatics, scope of bioinformatics and its applications, bioinformatics databases.
- Structural bioinformatics: Protein structure and its visualisation, comparison and classification, protein structure prediction, RNA structure prediction, compression of genomic sequences such as Burrows–Wheeler transform, etc.
- Pairwise sequence alignments and database search: Scoring matrix, Needleman-Wunsch algorithm, Smith-Waterman algorithm, Gotoh algorithm, heuristic methods
- Phylogenetic tree and multiple sequence alignment: Neighbour-joining and UPGMA algorithms, phylogenetic tree, Sequence profile & profile based alignments
- Pattern Recognition: Clustering and visualisation, Hidden Markov models and Viterbi algorithm
- Genomics and proteomics: Genome mapping, genome assembly, genome comparison, functional genomics, proteomics and metabolomics

Teaching/Learning Methods:

Lectures, Practical demonstration, recitation oral questions, vocabulary drills, and simulations.

Assessment Strategy:

- In-course Assessment (Theory) 10%
- In-course Assessment (Practical) 30%
- End-of-course Examination 60%

References:

- B. Bergeron, Bioinformatics Computing, Prentice Hall, 2002.
- K. Stephen, Introduction to Bioinformatics: A Theoretical and Practical Approach, 1st Ed, 2003.
- F. Azuaje and J. Dopazo, Data Analysis and Visualization in Genomics and Proteomics, John Wiley, 1st Ed, 2005

Course Code:	**CSC210S3**		
Course Title:	**Web Technologies**		
Credit Value:	**03**		
Core/Optional:	**Core**		
Hourly Breakdown:	**Theory**	**Practical**	**Independent Learning**
	15	60	75

Objectives:

Develop proficiency in designing web applications using different emerging technologies and best practices.

Intended Learning Outcomes:

- Design websites using advanced features of Markup and Client-side scripting languages
- Use XML technologies for web applications
- Employ knowledge on web programming to develop and maintain web applications
- Develop secure web-based systems using server-side scripting languages
- Recommend practices that ensure legal and ethical responsibilities

Course Contents:

- Advanced use of scripting languages: Client-side scripting (HTML, CSS, JavaScript, etc.) and Server-side scripting (PHP, JSP, ASP, etc.)
- XML Technologies: XSL, XSLT, xPath and xQuery
- Secure web programming: Authentication, access control, session management, SQL injections and cross site scripting (XSS)
- Trends in Web development: Web 2.0, AJAX, JSON, Web Services
- Best practices in Web Development: Architectural patterns, search engine optimization (SEO), frameworks, auditing and logging

Teaching/Learning Methods:

Lectures, practical demonstration, assignments, small group discussions, individual mini projects

Assessment Strategy:

- In-course Assessments (Theory) 10%
- In-course Assessments (Practical) 30%
- End-of-course practical Examination 60%

References:

- S. Purewal, Learning Web App development, 1st Ed., 2014.
- D. Stuttard and M. Pinto, The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws, 2nd Ed., 2011.
- J. J. Jackson, Web Technologies: A Computer Science Perspective, 1st Ed., 2006.
- A. Godbole and A. Kahate, Web Technologies, TCP/IP, Web/Java Programming, and Cloud Computing, McGraw Hill Education, 3rd Ed, 2017.

Course Code:	**CSC211S2**		
Course Title:	**Emerging Trends in Computer Science**		
Credit Value:	**02**		
Core/Optional:	**Core**		
Hourly Breakdown:	**Theory**	**Practical**	**Independent Learning**
	15	30	55

Objectives:

Provide an overview of the emerging trends in computer science.

Intended Learning Outcomes:

- Demonstrate familiarity with latest trends in computer science and their applications
- Describe the key architectures and applications in edge computing
- Summarise standard open-source cloud and edge computing software for data analytics
- Build microcontroller programs for IoT
- Discuss the latest languages and frameworks used in IT industries

Course Contents:

- Edge computing: Introduction to edge computing, cloud computing analytics pipeline, cloud databases
- Data analytics: Introduction to deep learning, data mining, and its applications; introduction to Hadoop, Spark, and MapReduce
- Internet of things (IoT): IoT concepts and technologies, its applications, micro-controller programming using sensors and actuators with arduino, IoT security and privacy issues
- Blockchain: Fundamentals of blockchain, distributed ledger technology, cryptocurrency, and related algorithms
- Introduction to mobile application development: Mobile app development platforms (Android, iOS, etc.), development and deployment of applications

[The course content will come directly from research papers, articles, and documentation of cloud and data center architectures and technologies.]

Teaching/Learning Methods:

Lectures, Guest lectures, TechTalks, workshops, industrial visit

Assessment Strategy:

- Formative Assessment: Industrial Visit*
- Summative Assessment: Individual/Group Assignments†

*Students will be taken to four to six leading software development companies in Sri Lanka in one or two industrial visit(s). Each visit may take up to three days. The type of assignments includes but are not limited to presentations and report writings on the observation of the industrial visit.

†At the end of each of the five chapters, students will be given five assignments (including programming tasks) based on the key areas covered in the five chapters. Of the five assignments, at most two may be done in groups.

Course Code:	**CSC212S2**		
Course Title:	**Professional Practice**		
Credit Value:	**02**		
Core/Optional:	**Core**		
Hourly Breakdown:	**Theory**	**Practical**	**Independent Learning**
	30	--	70

Objectives:

Provide a viewpoint on the commercial realities of software professionals and their required behavioural skills in day to day activities as an Information Technology professional.

Intended Learning Outcomes:

- Discuss the concepts of professional practice in computing
- Explain the context in which computer professionals work
- Apply the key skills, knowledge, attributes and attitudes required to be an IT professional, with particular reference to professional practice, code of ethics and professional standards
- Analyse legal issues in relation to data privacy and software use
- Recognize professional conduct in an ethical manner in day to day activities as an IT professional
- apply the principles of group work and reflect on the nature of working in teams, with the appreciation of the issues, such as ethics, conflict resolution, negotiation in culturally diverse workplace

Course Contents:

- Computer ethics and professional practice: Ethical argumentation and theories, moral assumptions and values, the role of computing professional, and professional communication practices
- Intellectual property: Intellectual property rights, Intangible digital intellectual property, legal foundations for intellectual property protection
- Privacy and data protection: Privacy of computer data, respecting human dignity, protecting data stored on computers, ethical hacking and its implications
- Security policies, laws and computer crimes: Computer crimes and legal redress for computer criminals, Issues surrounding the misuse of access and breaches in security, crime prevention strategies

Teaching/Learning Methods:

Lecture, small group discussions, tutorial classes

Assessment Strategy:

- In-course Assessments 30%
- End-of-course Examination 70%

References:

- G. W. Reynolds, Ethics in Information Technology, 5th Ed, 2014.
- M. F. Bott, Professional Issues in Information Technology, The British Computer Society, 2nd Ed, 2014.
- ACM Code of Ethics, ACM, www.acm.org, 2017.

Level–3S

Effective from the Academic Year: 2018/2019

Department of Computer Science
Faculty of Science
University of Jaffna
Sri Lanka

Developed in June 2019

Course Code:	**CSC301S3**		
Course Title:	**Rapid Application Development**		
Credit Value:	**03**		
Core/Optional:	**Core**		
Hourly Breakdown:	**Theory**	**Practical**	**Independent Learning**
	45	--	**105**

Objectives:

Provide knowledge, skills, and attitudes to rapidly develop software applications by choosing suitable approaches and best practices

Intended Learning Outcomes:

- Describe the concepts of software development methodologies
- Demonstrate the importance of Rapid Application Development (RAD) and its key elements
- Discuss how systems analysts interact with users, management, and other information systems professionals for gathering requirements
- Analyse the development lifecycle of a given software project
- Develop a software rapidly by best practices and tools

Course Contents:

- Introduction to RAD: Issues with traditional software development, advantages and disadvantages of RAD practices, pillars of RAD
- Key elements of RAD: Teamwork, risk management, project scheduling, project estimation
- Agile Software Development: Agile manifesto, agile methodologies such as SCRUM, extreme programming, Lean, and Kanban, Agile vs waterfall model
- Information Requirements Analysis: Determining system requirements, Interactive information gathering methods such as interviewing, Joint Application Development and Questionnaires. Unobtrusive information gathering methods such as sampling and investigation.
- Analysis Process: Data flow diagrams, analysing systems using data dictionaries, process specifications and structured decisions, designing effective input and output, designing databases
- Testing: Fundamentals of testing, black-box testing techniques, white box testing techniques, levels of testing, test cases
- Quality Assurance and Implementation: Ensuring data quality, six sigmas, quality assurance through software engineering, implementing information system, software testing process, evaluation techniques
- Best Practices and Tools: Software architectural patterns, software design patterns and software version control (SVC)
- Software Project Management: Work breakdown and cost estimation, Break-even analysis, cash-flow analysis, present value analysis, project scheduling using Gantt chart and PERT diagrams

Teaching and Learning Methods:

Lectures, Tutorial discussions, Case studies, Assignments, Guided Learning

Assessment Strategy:

- In-course Assessments 30%
- End-of-course Examination 70%

References:

- A. Stellman and J. Greene, Learning agile: Understanding Scrum, XP, lean, and kanban. O'Reilly, 2014.
- S. McConnell, Rapid development: Taming wild software schedules, Pearson Education, 1996.
- J. Loeliger, Version Control with Git, O'Reilly Media, 2012.
- E. Kendall and J. E. Kendall, System Analysis and Design, 9th Ed, Pearson, 2013.

Course Code:	**CSC302S2**		
Course Title:	**Computer Programming III**		
Credit Value:	**02**		
Core/Optional:	**Core**		
Hourly Breakdown:	**Theory**	**Practical**	**Independent Learning**
	--	**90**	**110**

Objectives:

Provide hands on practice in network socket programming, computer graphics, rapid application development, and network & server management

Intended Learning Outcomes:

- Apply rapid development methodologies used in the software industry
- Create software applications using development frameworks
- Practice application programming interface (API) for computer graphics
- Implement algorithms for computer graphics applications
- Write socket programming using Python libraries
- Setup web servers to enable interactions with other web servers using network protocols
- Demonstrate ability to configure, administer and secure local area network devices
- Administer Linux based systems

Course Contents:

- Version Control Systems: Introduction to GitHub and its workflow, branching, merging pull requests, working with teams on GitHub, creating task lists
- Development frameworks: Introduction to frameworks such as Laravel, setting up and install Laravel framework
- Software Applications: View/Session/Application management, databases in web application with Laravel
- Socket programming using Python: Client-server and TCP/UDP programming, multithreaded proxy server, reliable transport layer programs, distributed programs to implement routing algorithms, open and proprietary network applications development
- Network Design, Management and Troubleshooting: Setting up LAN, configuring and managing devices such as switches and routers with access controls, IP address configurations and troubleshooting
- Host Administration with LINUX: Basic commands, files, directories and file system, editors, processes, users and group management, package management, shell scripts
- OpenGL: Construction of interactive user interfaces, fundamentals of 2D and 3D graphics
- Computer Graphics Algorithms and Methods: Object modelling and representation, mapping and clipping, 2D and 3D transformations, rendering for visual realism

Teaching and Learning Methods:

Lectures, Use of multimedia presentations, Laboratory experiments, Tutorial discussions, Assignments

Assessment Strategy:

Semester-1		Semester-2	
● In-course Assessments (Practical)	20%	● In-course Assessments (Practical)	20%
● End-of-Semester Practical Examination	30 %	● End-of-Semester Practical Examination	30 %

References:

- M. Stauffer, Laravel: Up and Running: A Framework for Building Modern PHP Apps, O'Reilly Media, 2019.
- J. Kurose and K. Ross, Computer Networking: A Top-Down Approach, 7th Ed., Addison Wesley, 2017.
- S. Guha, Computer Graphics Through OpenGL: From theory to experiments 3rd Ed., CRC, 2019.
- W. Odom, CCNA Routing and Switching 200-125 Official Cert Guide Library, 1st Ed., Cisco Press, 2016.

Course Code:	**CSC303S2**		
Course Title:	**Data Communication and Computer Networks**		
Credit Value:	**02**		
Core/Optional:	**Core**		
Hourly Breakdown:	**Theory**	**Practical**	**Independent Learning**
	30	--	**70**

Objectives:

Provide in-depth understanding of architectures, algorithms, and standards in data communication

Intended Learning Outcomes:

- Describe data communication principles, layered architectures and protocols
- Discuss routing and switching principles, and algorithms
- Distinguish Local Area Network (LAN) standards, topologies, hardware and their selection criteria for enterprise usage
- Formulate network services and applications by taking into account of quality of service, scalability and maintenance

Course Contents:

- Fundamentals of Digital Communications: Design issues related to data transfer, compare and contrast the circuit and packet switching technologies, multiplexing techniques and error control mechanisms
- Network Architectures: Principles of layered architecture, roles of layers in the OSI and TCP/IP models
- Internet Protocols: Application layer protocols in the Internet, TCP/IP protocol suite, transport and network protocols with an emphasis on TCP/IP model, IP addressing and subnetting, troubleshooting in IP networks, IP routing mechanisms, IP versions 4 and 6, MAC layer and its protocols
- Routing and Switching: Routing and switching fundamentals, router architecture, routing algorithms, issues to consider in designing routing protocols
- LAN: Technologies such as IEEE 802 LAN standards, channel allocation in LAN segments and solutions, Ethernet and Ethernet networking using hubs and switches, problems associated in deploying wireless LANs and solutions

Teaching and Learning Methods:

Lectures, Recitation oral questions, Tutorial discussions, Supplementary reading

Assessment Strategy:

- In-course Assessments 30%
- End-of-course Examination 70%

References:

- J. F. Kurose and K. W. Ross, Data Communication and Computer Networks: A top-down approach, 7th Ed. Addison Wesley, 2017.
- A. S. Tanenbaum, and D. J. Wetherall, Computer Networks, 5th Ed., Pearson Education, 2011.
- L. L. Peterson and B. S. Davie, Computer Networks: A Systems Approach, 5th Ed., Morgan Kauffman, 2011.

Course Code:	**CSC304S3**		
Course Title:	**Team Software Project**		
Credit Value:	**03**		
Core/Optional:	**Core**		
Hourly Breakdown:	**Theory**	**Practical**	**Independent Learning**
	20	--	**280**

Objectives:

Provide an opportunity to improve the skills and knowledge of students to develop software as a team using software engineering principles

Intended Learning Outcomes:

- Apply software engineering principles and practices for the planning and development of a software product
- Practice as an effective player of a software project team
- Use appropriate tools, principles and best practices for developing an application
- Create professional-quality deliverables
- Develop an application based on a given set of requirements in order to deploy the application at the client site
- Demonstrate abilities to manage pressures and procedures of a team work in an industrial setup

Course Description:

- This course unit introduces and applies a range of topics in software engineering and rapid application development in the context of a team project
- Students will be assigned to a group of three to four members and each group works to specify, design, implement, and document a software project
- The course unit is oriented around directed and self-paced learning, supported by weekly mentoring and discussions

Teaching and Learning Methods:

Mentoring, Small group discussions, Case studies, Presentations, Demonstrations

Assessment Strategy:

- Team Software Project Report 30%
- Software product and deployment of the software 40%
- Project Presentation and individual viva-voce 30%

Course Code:	**CSC305S2**		
Course Title:	**Graphics and Visual Computing**		
Credit Value:	**02**		
Core/Optional:	**Core**		
Hourly Breakdown:	**Theory**	**Practical**	**Independent Learning**
	30	**--**	**70**

Objectives:

Provide in-depth knowledge in the core concepts of computer graphics including object modelling, transformations, and rendering

Intended Learning Outcomes:

- Discuss the fundamental concepts in computer graphics
- Describe the standard methods in object modelling and representation
- Apply transformation functions to animate 2D and 3D objects on view-planes
- Use rendering methods and algorithms to create photo-realistic interactive scenes from 2D and 3D models

Course Contents:

- Fundamental Concepts: Applications of computer graphics, image representations, vector vs. raster graphics, colour models
- Object Modelling and Representation: Rasterization of lines and circles, parametric forms of curves and surfaces, solid modelling with polygonal meshes
- Mapping and Clipping: Window to viewport mapping, algorithms for clipping lines, and polygons
- 2D and 3D Transformations: Affine transformations in 2D and 3D, coordinate transformations, view plane and view volume, projections, viewing transformation
- Basic rendering for visual realism: Visibility and occlusion (such as depth buffering, Painter's algorithm, and ray tracing), polygon filling, texture mapping, and shading models

Teaching and Learning Methods:

Lectures, Tutorial discussions, Assignments, Guided learning

Assessment Strategy:

- In-course Assessments 30%
- End-of-course Examination 70%

References:

- S. Marschner, and P. Shirley, Fundamentals of Computer Graphics, CRC Press, 4th Ed., 2015.
- S. Guha, Computer Graphics Through OpenGL: From Theory to Experiments 3rd Edition, CRC, 2019.
- D.D. Hearn, M.P. Baker, and W. Carithers, Computer Graphics with OpenGL, 4th Ed., 2010.
- S.J. Gortler, Foundations of 3D Computer Graphics, MIT Press, 2012.

Course Code:	**CSC306S3**		
Course Title:	**Advanced Database Design and Systems**		
Credit Value:	**03**		
Core/Optional:	**Core**		
Hourly Breakdown:	**Theory**	**Practical**	**Independent Learning**
	45	**--**	**105**

Objectives:

Provide in-depth understanding of the design, implementation and administration features of database management systems to effectively develop, and manage medium to large-scale databases

Intended Learning Outcomes:

- Describe the concepts of database & distributed database design, and their logical & physical organisations
- 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 Contents:

- Relational Modelling: Concepts of data modelling, enhanced entity-relationship(EER) model, use of unified modelling language (UML), higher level normalisation
- Physical Organisation of Databases: Storage and file structure, indexing, database efficiency and tuning
- Query Optimisation: Factors governing query optimization, centralized query optimization
- Transaction: Transaction processing, concurrency control, recovery techniques
- Distributed Database Management Systems: Data fragmentation, replication and allocation, transaction processing, concurrency control and recovery in distributed databases
- Advanced DBMS Concepts: Advanced technologies in DBMS and enhanced data models

Teaching and Learning Methods:

Lectures, Tutorial discussions, Assignments, Guided learning

Assessment Strategy:

- In-course Assessments 30%
- End-of-course Examination 70%

References:

- R. Elmasri and S.B. Navathe, Fundamentals of Database Systems, 7th Ed., Addison-Wesley, 2015.
- C.J. Date, An Introduction to Database Systems, 8th Ed., Addison-Wesley, 2003.
- Ramakrishnan and Gehrke, Database Management Systems, 3rd Ed., McGraw-Hill, 2003.

Course Code:	**CSC307S3**		
Course Title:	**Advanced Topics in Computer Networks**		
Credit Value:	**03**		
Core/Optional:	**Core**		
Hourly Breakdown:	**Theory**	**Practical**	**Independent Learning**
	45	**--**	**105**

Objectives:

Provide in-depth knowledge in advanced and emerging trends in network virtualisation and software defined networks

Intended Learning Outcomes:

- Demonstrate a deeper understanding of modern computer networks, applications, and network services
- Explain how different networking technologies at the same or different layers interact and affect each other in a large-scale system
- Appraise network technologies with respect to system requirements, based on information from recent research and technical documentation
- Perceive trends in large scale networks such as virtualization and software defined networking capability
- Evaluate network technologies, applications, and services through simulation and emulation experiments

Course Contents:

- Routing in the Internet: Intra and inter-domain routing, unicast and multicast routing protocols, traffic classes and measurements, multi-protocol label switching(MPLS), storage area networks (SAN), data and the control planes
- Fundamental Properties of Computer Networks: Congestion control, queueing and scheduling, quality of service, quality of experience, Power laws
- Software Defined Networks: Distinguish between traditional networks and software defined networks, SDN origins and evolution, centralized and distributed control and data planes, open flow protocol, SDN controllers, Mininet, NOX/POX, network programming using SDNs
- Network Virtualization: Introduction to network virtualization, constructing virtual network topologies on top of physical network topologies, virtual machines, architectural issues
- Network Measurement: Measurement, modelling and analysis methods using real network data, Wireshark tool to monitor active networks, network simulations and emulations

Teaching and Learning Methods:

Lectures, Recitation of oral questions, Supplementary reading, Practical demonstration

Assessment Strategy:

- In-course Assessments 30%
- End-of-course Examination 70%

References:

- J. F. Kurose and K. W. Ross, Data Communication and Computer Networks: A top-down approach, 7th Ed. Addison Wesley, 2017.

- A. S. Tanenbaum, and D. J. Wetherall, Computer Networks, 5th Ed., Pearson Education, 2011.
- L. L. Peterson and B. S. Davie, Computer Networks A Systems Approach, 5th Ed., Morgan Kauffman, 2011.

Course Code:	**CSC308S3**		
Course Title:	**Artificial Intelligence**		
Credit Value:	**03**		
Core/Optional:	**Core**		
Hourly Breakdown:	**Theory**	**Practical**	**Independent Learning**
	30	**30**	**140**

Objectives:

Provide in-depth knowledge on design and analysis of intelligent systems for solving problems that are difficult or impractical to resolve using traditional approaches

Intended Learning Outcomes:

- Formulate an efficient Intelligent system model for a problem expressed in natural language
- Use knowledge representation for theorem proving based on resolution procedure
- Apply appropriate uninformed, informed or local search algorithms for solving problems
- Develop logic programs with the significance of language semantics
- Devise a plan of action to achieve a goal using standard AI methods
- Illustrate the working of natural language processing techniques

Course Contents:

- Introduction: Practical examples of artificial intelligence, intelligent agents, environments, intelligent behaviour, rational behaviour & Turing test
- Problem Solving by Searching: Problem-solving agents, uninformed search strategies, informed (Heuristic) search strategies
- Local Search and Optimization Algorithms: Hill climbing search, simulated annealing, local beam search, genetic algorithms, searching in different environments, adversarial search
- Planning: Classical planning, planning as state-space search
- Knowledge Representation: Horn clause, resolution, theorem proving, ontology engineering, representing objects and events
- Natural Language Processing: Language models, text classification, information retrieval, information extraction

Teaching and Learning Methods:

Lectures, Tutorial discussions, Guided learning, Assignments

Assessment Strategy:

- In-course Assessment (Theory) 15%
- In-course Assessment (Practical) 15%
- End-of-course Examination 70%

References:

- S. J. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 3rd Ed., Prentice Hall, 2010.
- G.F. Luger, Artificial Intelligence - Structures and Strategies for Complex Problem Solving, 6th Ed., Pearson & Addison Wesley, 2009.
- P. H. Winston, Artificial Intelligence, 1st Ed., Addison Wesley, 1993.

Course Code:	**CSC309S3**		
Course Title:	**High Performance Computing**		
Credit Value:	**03**		
Core/Optional:	**Core**		
Hourly Breakdown:	**Theory**	**Practical**	**Independent Learning**
	30	**30**	**140**

Objectives:

Provide in-depth knowledge on the computational aspects of high performance computing and methods of parallel programming

Intended Learning Outcomes:

- Discuss basics of high performance computing and their usage
- Describe different parallel architectures, interconnection networks
- Transform sequential algorithms into efficient parallel algorithms
- Devise parallel programming models and parallel algorithms for solving computational problems
- Analyse parallel programming paradigms and their semantics and correctness issues
- Assess parallel algorithms based on their complexity and scalability

Course Contents:

- Introduction to High Performance Computing: Cluster computing, grid computing, cloud computing, parallel & distributed computing, fault tolerance, concurrency, nondeterminism, locality
- Parallel Architectures: Taxonomy, data versus control parallelism (SIMD/Vector, pipelines, MIMD, multi-core, heterogeneous), shared versus distributed memory, interconnection networks for parallel computers
- Scheduling and Analytical Modelling: Cost of computation and scalability, model-based notions, handling scheduling issues
- Parallel Algorithms: Communication operations, algorithmic paradigms (Divide and conquer, recursion, Series, parallel composition), computation on matrices, sorting, graph algorithms, search algorithms
- Concepts of Parallel Programming: Distributed-memory programming (MPI), shared-memory programming (OpenMP, CUDA)

Teaching and Learning Methods:

Lectures, Practical demonstration, Assessments, Tutorial discussions, Guided learning

Assessment Strategy:

- In-course Assessment (Theory) 15%
- In-course Assessment (Practical) 15%
- End-of-course Examination 70%

References:

- J.C. Zbigniew, Introduction to Parallel Computing, 1st Ed., Cambridge University Press, 2017.
- J. Sanders, and E. Kandrot, CUDA by Example: An Introduction to General-Purpose GPU Programming, 1st Ed., Addison-Wesley Professional, 2010.
- A. Grama, A. Gupta, G. Karypis, and V. Kumar. “Introduction to Parallel Computing”, 2nd Ed., Addison-Wesley, 2003.

Course Code:	**CSC310S3**		
Course Title:	**Image Processing and Computer Vision**		
Credit Value:	**03**		
Core/Optional:	**Core**		
Hourly Breakdown:	**Theory**	**Practical**	**Independent Learning**
	30	**30**	**140**

Objectives:

Provide in-depth knowledge in image processing and computer vision techniques to solve real-world problems, and develop skills for research in these fields

Intended Learning Outcomes:

- Describe the basic concepts of image processing and computer vision
- Perform visual tasks in sequences of image analysis operations, representations, specific algorithms, and inference principles
- Explain image processing techniques in the spatial and frequency domain
- Analyse a range of algorithms for image processing and computer vision
- Develop basic computer vision algorithms for image retrieval and image recognition
- Apply image processing and computer vision techniques to solve real-world problems

Course Contents:

- Digital Image Fundamentals: Image representation, sampling and quantisation, image size, resolution, pixel neighbours, connectivity, arithmetic and logical operations for images
- Image Enhancement in Spatial Domain: Intensity transformations, histogram equalization and specification, filter operations for smoothing, sharpening images, and noise reduction
- Image Enhancement in Frequency Domain: The Fourier transform and its properties, Fast Fourier Transform (FFT), filter operations for smoothing, sharpening images, and noise reduction
- Morphological Image Processing: Dilation and erosion, opening and closing, basic morphological applications
- Image Segmentation: Thresholding, edge detection, region growing
- Introduction to Computer Vision and its Applications: Human eye-brain system as a model for computer vision, biometric applications, automated navigation
- Introduction to Object Recognition: Feature types and descriptors, template matching, bag-of-features framework, feature matching, convolutional neural networks (CNNs)

Teaching and Learning Methods:

Lectures, Assignments, Poster presentation, Guided learning

Assessment Strategy:

- In-course Assessments (Theory) 15%
- In-course Assessments (Practical) 15%
- End-of-course Examination 70%

References:

- W. Burger and M.J. Burge, Principles of Digital Image Processing: Fundamental Techniques, Springer, 3rd Ed., 2009.
- M. Sonka, R. Boyle and V. Hlavac, Image Processing, Analysis and Machine Vision, 3rd Ed., Springer, 2008.
- R.C. Gonzalez and R.E. Woods, Digital Image Processing, 3rd Ed., Pearson, 2007.
- L.G. Shapiro and G. Stockman, Computer Vision, Prentice Hall, 2001.

Course Code:	**CSC311S3**		
Course Title:	**Machine Learning**		
Credit Value:	**03**		
Core/Optional:	**Core**		
Hourly Breakdown:	**Theory**	**Practical**	**Independent Learning**
	30	**30**	**140**

Objectives:

Provide knowledge on the concepts of machine learning techniques for data analysis and modelling

Intended Learning Outcomes:

- Describe a range of supervised, unsupervised and reinforcement learning algorithms
- Explain different deep learning techniques
- Perform pre-processing operations on data to mine useful information
- Identify appropriate learning paradigms for given data mining problems
- Apply machine learning algorithms on data to identify new patterns or concepts

Course Contents:

- Introduction to Machine Learning: Machine intelligence and applications, concepts, instances, attributes and their types, and handling sparse data, missing & inaccurate values in data, handling categorical data
- Supervised Learning: Introduction to classification and regression, rule-based learning, decision tree learning, Naive Bayes, k-nearest neighbour, support vector machines, neural networks, linear regressions, introduction to boosting
- Unsupervised Learning: K-means clustering, Gaussian mixture models (GMMs), hierarchical clustering Reinforcement Learning: Markov decision processes (MDP), value functions, returns and value functions, Bellman equation and optimality
- Introduction to Deep Learning: Convolutional neural network (CNN), Recurrent neural network (RNN)
- Dimensionality Reduction: PCA, feature selection
- Experimental Setup and Evaluation: Training and testing, cross-validation, confusion matrices and evaluation measures such as accuracies, mean square errors, ROC values

Teaching and Learning Methods:

Lectures, Vocabulary drills, Assignments, Laboratory experiments, Guided learning

Assessment Strategy:

- In-course Assessment (Theory) 15%
- In-course Assessment (Practical) 15%
- End-of-course Examination 70%

References:

- C.M. Bishop, "Pattern Recognition and Machine Learning", 2007.
- R.O. Duda, P.E. Hart, D.G. Stork, "Pattern Classification", 2nd Ed., Wiley, 2000.
- T. Mitchell, "Machine Learning", McGraw Hill, 1997.
- I.H. Witten, E. Frank, M.A. Hall, "Data Mining: Practical Machine Learning Tools and Techniques", 3rd Ed., Morgan Kaufmann Series, 2011.

Course Code:	**CSC312S3**		
Course Title:	**Mobile Computing**		
Credit Value:	**03**		
Core/Optional:	**Core**		
Hourly Breakdown:	**Theory**	**Practical**	**Independent Learning**
	45	--	**105**

Objectives:

Provide in-depth understanding of the concepts in mobile computing and the state of the art trends in mobile computing research

Intended Learning Outcomes:

- Describe the concepts of mobile wireless communications
- Discuss realistic problems in wireless communication
- Identify latest research trends in mobile computing
- Apply knowledge for mobile applications development
- Appraise routing and forwarding protocols for mobile ad hoc networks
- Recommend ad-hoc network based solutions for real world problems

Course Contents:

- Overview of Wireless Networks: Wireless communication properties, wireless impairments, multiplexing in wireless communications, the need for a specialized MAC
- Routing in Wireless Networks: issues in routing for wireless networks, wireless routing protocols
- Ad-hoc Networks: mobile ad-hoc networks, sensor networks and vehicular ad-hoc networks, routing in ad-hoc networks, latest trends in ad-hoc networking
- Mobile Applications Development Environments: mobile platforms, development and deployment of applications
- Common Paradigms in Mobile Computing: low power computing, mobile computing in resource constrained environments, fault tolerance, and persistence

Teaching and Learning Methods:

Lectures, Assignments, Tutorial discussions, Guided learning

Assessment Strategy:

- In-course Assessments 30%
- End-of-course Examination 70%

References:

- J. Schiller, "Mobile Communications", 2nd Ed., Addison Wesley publishers, 2004.
- M. Yener and O. Dundar, "Expert Android Studio", 1st Ed., Wrox publications, 2016.

Level–4S

Effective from the Academic Year: 2019/2020

Department of Computer Science

Faculty of Science

University of Jaffna

Sri Lanka

Developed in December 2020

.

Course Code:	CSC401S3		
Course Title:	Advanced Algorithms		
Credit Value:	**03**		
Hourly Breakdown:	**Theory**	**Practical**	**Independent Learning**
	45	--	**105**

Objectives:

Provide in-depth knowledge for designing efficient algorithms using appropriate data structures and a variety of advanced computational techniques.

Intended Learning Outcomes:

- Perform algorithm analysis using appropriate techniques
- Apply algorithms for solving problems of various complexity
- Identify suitable data structure(s) for efficient problem solving
- Formulate a real world problem into a model solvable by standard algorithmic approaches
- Defend NP-Hard & NP-Complete problems

Course Contents:

- Analysis of algorithms: Order of growth and asymptotic notation, master method, amortised analysis
- Algorithms for sorting and order statistics: Standard sorting algorithms, sorting in linear time, medians, order statistics
- Advanced data structures: Hash table, binary heap, binary search tree, red-black tree, augmenting data structures, B-trees, Fibonacci heaps, utilising data structures for problem solving.
- Dynamic programming: Elements of dynamic programming (optimal substructure, overlapping sub-problems, memoization), application of dynamic programming approach for problem solving
- Greedy algorithms: Elements of Greedy strategy (optimal substructure, overlapping sub-problems, Greedy strategy), comparison of dynamic programming and Greedy approach, application of Greedy approach for problem solving
- P, NP, NP-Hard and NP-Completeness: Introduction to class of problems, NP-completeness (Cook's theorem), classic NP-complete problems, reduction techniques

Teaching and Learning Methods:

Lectures, Tutorial discussion, e-based teaching, Open Educational Resources, Assignments, Guided learning

Assessment Strategy

- In-Course Assessments 30%
- End-of-Course Examination 70%

References:

- Tardos, J. K. E., Algorithm Design, 2nd Ed., Pearson Education, 2011.
- Sedgewick, R., Algorithms, 4th Ed., Addison-Wesley Professional, 2011.
- Thomas, H. Cormen, T. Leiserson, C. E., Rivest, R. L., and Stein, C., Introduction to Algorithms, 3rd Ed., MIT Press, 2009.

Course Code:	CSC402S3		
Course Title:	Compiler Design		
Credit Value:	**03**		
Hourly Breakdown:	**Theory**	**Practical**	**Independent Learning**
	45	--	**105**

Objectives:

Provide in-depth knowledge of compiler components and principles involved in compiler design.

Intended Learning Outcomes:

- Discuss the functionalities of a compiler
- Represent a specified language using nondeterministic and deterministic finite automata
- Analyse the syntax of a language using top-down and bottom-up parsing approaches
- Discuss implementation techniques for a compiler
- Design algorithmic routines for type checking and code generation

Course Contents:

- Introduction: Compilers, Interpreters, and phases of compilation process
- Lexical Analysis: Regular expressions and their properties, converting regular expressions to Nondeterministic Finite Automata (NFA), Converting NFA to Deterministic finite automata (DFA), Eliminating dead states and minimising DFA
- Syntax Analysis: Context-free grammars, Syntax trees, Ambiguity in Grammar, Operator precedence, eliminating left-recursion, Left-factorisation, Top-down parsing methods, Bottom-up parsing methods, Resolving conflicts in parsing tables
- Scope and Symbol Tables: Dealing with scopes of identifiers, Efficient implementations of symbol tables
- Type Checking: The design space of types, Attributes, Type checking procedures
- Code Generation: Syntax Directed Translations, Intermediate-Code Generation, Machine-Code Generation

Teaching and Learning Methods:

Lectures, Tutorial discussion, e-based teaching-learning, Open Educational Resources, Assignments, Guided Learning

Assessment Strategy

- In-Course Assessments 30%
- End-of-Course Examination 70%

References:

- Mogensen, T. E., Introduction to Compiler Design, 2nd Ed., Springer, 2017.
- Lam, M. S., Sethi, R., and Aho, A. V., Compilers: Principles, Techniques, and Tools, 2nd Ed., Pearson Education, 2013.
- Galles, D., Modern Compiler Design, 2nd Ed., Pearson Education, 2009.

Course Code:	CSC403S3		
Course Title:	Data Science		
Credit Value:	03		
Hourly Breakdown:	**Theory**	**Practical**	**Independent Learning**
	30	**30**	**90**

Objectives:

Provide theoretical and practical knowledge on data science for solving data-driven problems and improving research skills in data science.

Intended Learning Outcomes:

- Discuss fundamental concepts of linear algebra in relation to data science
- Analyse data-driven problems using probability and statistics
- Apply mathematical optimization techniques for solving data-driven problems
- Build neural networks using backpropagation algorithm
- Transform data-driven problems into computer programs
- Analyse data through visualization

Course Contents:

- Linear algebra: Overview of scalars, vectors, matrices, tensors, multiplication of matrices and vectors, norms, trace, rank, eigenvalues, and eigenvectors
- Probability and statistics: Overview of probability, marginal and conditional probabilities, independence and conditional independence, probability density functions, expectation, variance and covariance, and Bayes rule; Bayesian classifier, principal component analysis (PCA), linear discriminant analysis (LDA).
- Information theory: Entropy, cross-entropy, KL divergence, mutual information
- Multivariate calculus and mathematical optimization for parameter estimation: partial derivatives, chain rule, gradient based optimization (Gradient Descent, Stochastic Gradient Descent), Jacobian and Hessian matrices, constrained optimization, neural networks and back propagation algorithm.
- Applications: Eigen faces, LDA based classification, Linear Regression, Least square minimization
- Emerging Technologies in Big Data Analytics: Usage of Open source frameworks (e.g. Hadoop).

Teaching and Learning Methods:

Lectures, Tutorials, Laboratory experiments, e-based teaching-learning, take home exercises, Simulations, Use of Open Educational Resources, Guided Learning

Assessment Strategy
● In-course Assessment (Practical) 30% ● End-of-course Examination 70%
References:
● Goodfellow, I., Bengio, Y. and Courville, A., Deep Learning, 1st Ed, MIT Press, 2016. ● Bishop, C.M., Pattern Recognition and Machine Learning, 1st Ed, Springer, 2006. ● Nielsen, M., Neural Networks and Deep Learning, Determination Press, 2019. ● Aggarwal, C. C., Linear Algebra and Optimization for Machine Learning, 1st Ed, Springer Nature, 2020

Course Code:	CSC404S3		
Course Title:	Information Systems Security		
Credit Value:	**03**		
Hourly Breakdown:	**Theory**	**Practical**	**Independent Learning**
	45	--	**105**

Objectives:

Provide knowledge to identify various security threats and propose suitable approaches to protecting Information Systems.

Intended Learning Outcomes:

- Identify various security threats and attacks on Information Systems.
- Explain security design principles.
- Elaborate techniques for data protection on Information Systems.
- Understand the techniques used to protect Computer Networks
- Explain the protocols that are utilised to protect the network and application layers of the Internet.

Course Contents:

- Introduction: Key security concepts, Critical characteristics of Information System, Secure-System life cycle, Security Professionals and the Organization.
- The need for security: Threats, Attacks, Secure software development.
- Security technologies: Firewalls, Virtual private networks, Intrusion detection and prevention systems, other security tools.
- Cryptography: Classical encryption techniques, Block ciphers, Data encryption standard, Advanced encryption standard, Public-Key cryptosystems.
- Cryptographic data integrity algorithms: Cryptographic hash functions, Message authentication codes, Digital signatures.
- Network security: Network access control and cloud security, Transport-level security, Wireless network security.
- Internet security: Internet Protocol security, web security (S-HTTP), email threats and email security (S/MIME)

Teaching and Learning Methods:

Lectures, e-based teaching-learning, Tutorial discussion, Assignments, Simulations, Use of Open Educational Resources, Guided Learning

Assessment Strategy

- In-Course Assessments 30%
- End-of-Course Examination 70%

References:

- Whitman, M. E. and Mattord, H. J., "Principles of Information Security", 6th Ed., Cengage Learning, 2017.
- Stallings, W., "Cryptography and Network Security: Principles and Practice", 4th Ed, Pearson Education Limited, 2017

- Paar, C., Pelzl, J., "Understanding Cryptography: A Textbook for Students and Practitioners" 1st Ed, Springer, 2014.

Course Code:	CSC405S3		
Course Title:	Systems and Network Administration		
Credit Value:	**03**		
Hourly Breakdown:	**Theory**	**Practical**	**Independent Learning**
	15	60	**75**

Objectives:

Provide theoretical and practical knowledge required to manage and maintain hosts, network connectivity devices, and various networked servers.

Intended Learning Outcomes:

- Administer computer systems and connectivity devices
- Configure systems and devices for different networking scenarios
- Design a data communication network required for a small/medium organisation
- Demonstrate ability in network and server management
- Implement security policies in networked systems

Course Contents:

- Host Management: Host hardware and maintenance, Basic commands, Files, Directories and File System, Editors, Processes, Users and group management, Package management, Automating system administration.
- Network Design and Management: Network connectivity devices, Host network configuration, Routing and Router configuration, IP addressing, subnetting, Switch configuration, Wireless equipment, VLAN, Inter-VLAN routing,
- Server Management: Install and manage server operating systems, Web server, e-mail server, Proxy server, DNS server, Content Servers,
- Security Policy Implementation: Firewall configuration, IP tables, Secure remote administration, Simple Intrusion Detection Techniques, Snort.

Teaching and Learning Methods:

Lectures, Practical, e-based teaching-learning, Open Educational Resources, Assignments, Online based training, Simulation, Guided Learning

Assessment Strategy

- In-Course Assessment (Theory) 20%
- Group Project 20%

- End-of-Course Examination (Practical) 60%

References:

- Blokdyk, G., "Computer Network Administration: A Clear and Concise Reference", 1st Ed, 5 Star Cooks, 2019
- Frisc, A., "Essential System Administration", 3rd Ed, O'Reilly Media Inc., 2002.
- Odom, W., CCNA 200-301 Official Cert Guide Library, 1st Ed, Cisco Press, 2019
- Bauer, M. D., "Linux Server Security", 2nd Edition, O'Reilly Media Inc., 2005

Course Code:	CSC406S6		
Course Title:	Research Project		
Credit Value:	**06**		
Hourly Breakdown:	**Mentoring**	**Practical**	**Independent Learning**
	20	--	580

Objectives:

Develop capability of carrying out scientific research in the computing domain for solving real world problems.

Intended Learning Outcomes:

- Identify a hypothesis and/or a research problem
- Formulate the detailed problem statement
- Frame a solution with appropriate research methodology
- Validate the proposed solution
- Perform scientific communication
- Defend the performed research, results and findings

Course Outline:

- Engage in an academic year long research project under the guidance of academic supervisor(s) with optional guidance by an external mentor
- Develop a research proposal with adequate literature review
- Carry out the research using appropriate research methodology
- Document and present the research

Teaching and Learning Methods:

Reading assignments in journals, Research seminars, Open Educational Resources, Documentation

Assessment Strategy

- Presentation of research proposal 20%
- Proposed solution and Implementation 40%
- Project Diary 10%

- Project report 10%
- Viva voce 10%
- Submission of abstract/poster/paper to a scientific forum 10%

Course Code:	CSC407S6		
Course Title:	Industrial Training		
Credit Value:	**06**		
Hourly Breakdown:	**Mentoring**	**Practical**	**Independent Learning**
	20	--	580

Objectives:

Provide an opportunity to develop skills and attitude, and gain experience in finding IT solutions to problems in an industrial environment.

Intended Learning Outcomes:

- Apply acquired knowledge in industrial environment
- Develop interpersonal, communication, management and team working skills
- Adapt to work readily in real industrial projects
- Perceive state-of-the-art industrial technologies

Course Outline:

- The industrial training is offered during the second semester.
- Students will be trained in an appropriate industry for a period of four to six months which amounts to 600 notional hours under the guidance of academic and industrial mentors.
- Any additional stay at the industry will not carry any additional credits.
- It is the student's responsibility to find a placement in consultation with the department. The department may assist the student in finding a placement in an appropriate industry.
- Students shall maintain a journal to record their progress activity during the training.
- Academic staff will visit the training institution at least once during the training period to monitor their progress.
- On completion of the industrial training, each student shall submit the report, journal and deliver an oral presentation.
- Students who fail to obtain a minimum grade of D+ in industrial training may opt for a general degree in Computer Science as the training is non-repeatable.

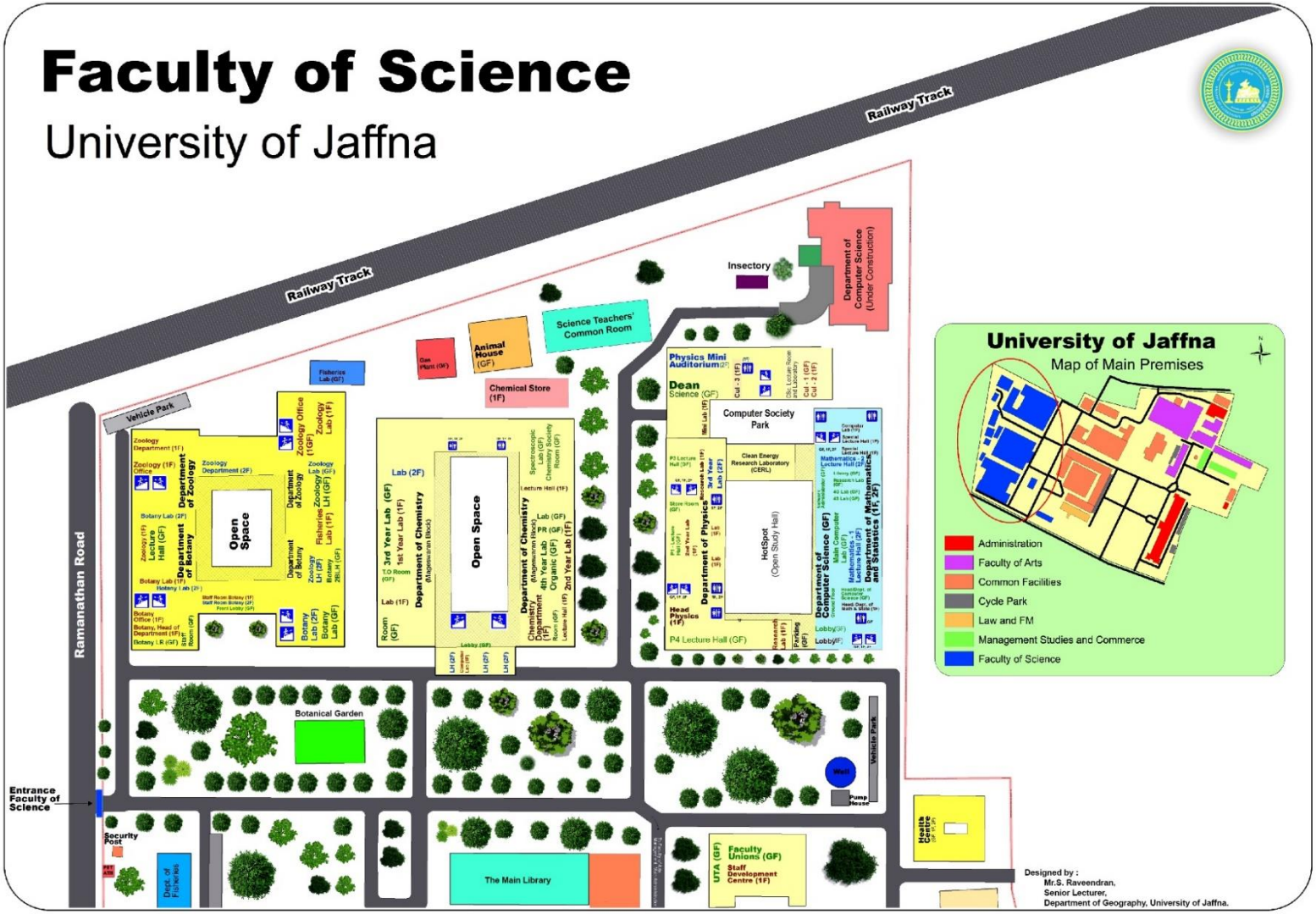
Training Methods:

Mentoring, Weekly recording of training diaries, Code reviews, Progress meetings, Supervised study

Assessment Strategy

- Training journal 20%
- Progress as per feedback from mentor(s) 20%
- Final Report 20%
- Presentation 40%

MAP OF THE FACULTY OF SCIENCE



Disclaimer

This study prospectus issued by the Department of Computer Science (DCS), University of Jaffna is compiled with information received up to October 2021. It is hereby informed that this study prospectus is only for general information. Any information contained herein should be confirmed by reference to the relevant authority. For additional information please refer to the Undergraduate Student Handbook issued by the Dean's Office, Faculty of Science, University of Jaffna.

Department of Computer Science

Faculty of Science

University of Jaffna

Jaffna

Sri Lanka.

Telephone: +94(0) 21 221 8194

Email: dcs@univ.jfn.ac.lk

Web: www.csc.jfn.ac.lk