DCS
UNIVERSITY OF JAFFNA

BSc (Computer Science)

(For Direct-intake Computer Science Students)

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:	CSC3015S3		
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.