POSTGRADUATE INSTITUTE OF SCIENCE

UNIVERSITY OF PERADENIYA

M.Sc. in Computer Science
2011/2012

1. INTRODUCTION

Nowadays, most of the science degree students require knowledge of Computer Science in order to find a suitable job. On the other hand, those who have done Computer Science as a subject during the undergraduate programme are looking for a competitive job. Information technology will exert a great influence in all science subjects in the future and therefore it is essential to train graduates for the next millennium. Computer science will be the first as well as the most innovative discipline that can interact with any other discipline in order to develop subjects, which have areas of interaction.

Today, the knowledge of computer science is a must for undergraduate as well as postgraduate degree students. All the enterprises (institutes, organizations, and companies) should have at least one qualified computer scientist. In this decade, most institutes need to develop by introducing or/and improving computer science in their curricula to suit actual demands. The postgraduate degree programme envisaged will give a comprehensive knowledge of recent issues in computer science.

2. OBJECTIVES OF THE PROGRAMME

The objective of this programme is to provide computer science/non-computer science graduates, advanced and enhanced knowledge of recent issues of information technology. At the completion of this course, the candidate will be able to fit into any computerized enterprise or a research institute.

3. PROGRAMME ELIGIBILITY

Applicants must possess a science-based degree (e.g. Physical/Biological/Engineering related degree), any other degree where a basic knowledge of Calculus, Number Theory and Algebra have been obtained or an equivalent qualification acceptable to the Postgraduate Institute of Science. Graduates, who have no basic knowledge in computer science, are expected to follow fundamentals of computer science (None credit courses). Depending on the courses followed at the degree level and on the recommendation of the advisor a candidate may be exempted from some of the preliminary courses.

4. PROGRAMME FEE

	M.Sc. programme fee
local candidates	Rs.110,000/-
SARRC countries	US $ 5,6000/-
other countries	US $ 11,200/-

Programme fees shall be paid in two instalments *(50% at the registration and the balance 50% within six months from registration)*. Other payments including registration fee, medical fee, library subscription, examination fee and deposits (science and library) should be paid according to the procedure stipulated by the PGIS.

5. THE PROGRAMME STRUCTURE AND DURATION

The programme shall be conducted on a course unit basis according to the common format developed by the Postgraduate Institute of Science. This is a full-time programme consists of course work and a research project.

Course work will be conducted over a period of two semesters of 15 - weeks each *(during weekends and/or weekdays)*. Satisfactory completion of a minimum of 24 credit units of course work is required for the programme. Students who are eligible to proceed to the M.Sc. degree programme are required to complete a minimum of 30 credit units, inclusive of 6 credit units allocated for a full-time research project (equivalent to six credit units) of a minimum of three months duration. Continuous attendance is compulsory during the period of research work.

Each candidate will be assigned an academic advisor, whose advice should be sought when planning the M. Sc. programme. The approval of the programme coordinator is necessary prior to the commencement of the programme. English will be the medium of instruction.

Programme Summary

Course Code	Course	Lecture hrs.	Practical hrs.	No. of Credits
Preliminary Courses				
SC 411	Introduction to Computer Science	30	-	-
SC 412	Introduction to theory of computation	30	-	-
SC 413	Data structures and software principles	30	-	-
SC 414	Introduction to Computer Architecture	30	-	-
SC 415	Programming and electronics laboratories	-	45	-
SC 416	Seminar	-	15	-
Semester I				
SC 531	Database systems *	30	-	2
SC 532	Combinatorial mathematics *	30	-	2
SC 533	Introduction to parallel computing *	30	-	2
SC 534	Programming language design and compilers *	30	-	2
SC 535	Operating system design **	30	-	2
SC 536	Graph theory*	30	-	2
SC 537	Computer Networks & Distributed systems **	30	-	2
SC 538	Artificial Intelligence **	30	-	2
SC 539	Advanced topics in computer graphics **	30	-	2
Semester II				
SC 546	Software engineering **	30	-	2
SC 547	Computer architecture *	30	-	2
SC 548	Systems analysis/Systems engineering *	30	-	2
SC 549	Artificial Neural Networks *	30	-	2
SC 550	Linear programming *	30	-	2
SC 551	Communication networks for computers *	30	-	2
SC 552	Digital image processing*	30	-	2
SC 553	Project management*	30	-	2
SC 554	Special topics in Computer Science **	30	-	2
SC 555	Laboratory work **	-	60	2
SC 597	Seminar **			1
SC 599	Research project (minimum of three months duration) **			6

Preliminary courses are not considered in the computation of the GPA

* *Optional Courses*

** *Compulsory Courses*

6. PROGRAMME CONTENTS

SC 411: Introduction to Computer Science *(30 hrs, no credits)*

Introduction and overview, Overview of the computer science curriculum, Intelligent machines and systems applications, Chemical, biological, and medical applications, Environmental and ecological applications, Information and educational applications, Engineering and scientific applications, Business and management applications, Communications and media applications.

Introduction to Computer Programming: Basic concepts, Basic components of programming languages; Variables, declarations, binding, procedures, functions, Simple algorithms operating on nonstructured data, Modularity in computer programming; Basics of constructing larger programs: abstraction and instantiation of program components, Structured data; Lists, stacks, queues, ordered binary trees.

Storing and accessing data structures, Operating on mutable data; Working with mutable data, object-based programming, data encapsulation. (Sections are covered using C, C^{++} and JAVA)

Textbook: Manis & Little, The Schematics of Computation, Prentice-Hall, 1995.
Laboratory work: Computer programming in C. Students are also required to write a weekly essay on the course material.

SC 412: Introduction to theory of computation *(30 hrs, no credits)*
Sets, Propositions, Two-valued Boolean algebra, Inductions, Recursion, Relations and functions, Graphs, Basic Techniques; Mathematical proofs, induction and recursion, gcd, Fibonacci numbers, Lame's theorem, Counting; Rules of sum and product, permutations, combinations. Pascal's triangle, binomial theorem, summation of binomial coefficients, Probability; Probability, inclusion/exclusion, conditional probability, analysis of expected behavior of algorithms, Recurrences; Method of operators/annihilators, divide and conquer recurrences/algorithms, Basic Algorithmic Techniques; Dynamic programming, greedy heuristics, Graphs and Trees; Definitions, Hamiltonian paths and Ore's theorem, depth first search and applications, Eulerian paths, breadth first search and applications, planarity/Platonic solids, Automata and Languages; Finite state machines, regular languages/closure properties, pumping lemma, context free languages/pumping lemma/BNF, Turing machines/computability.

Textbook: Kenneth Rosen, Discrete Mathematics and Its Applications, 3rd edition, McGraw-Hill, 1990, Cormen, Leiserson, and Rivest, Introduction to Algorithms, McGraw-Hill.
Laboratory work: None.

SC 413: Data structures and software principles *(30 hrs, no credits)*
Introduction, Program design concepts, Abstract data types, Basic data structures, Abstract data types for sets: operations and implementations, Sorting, Memory management, Graph algorithms, String algorithms, Arrays, records, pointers, indices, Recursion, Timing comparisons, Memory comparisons, Lists; Implementation: array/linked; ordered/unordered, Searching: introduction to set abstract data type, Stacks and queues, Trees; Pointer implementation; traversal, Binary search trees; Definition, Searching, Creation and insertion, Good and bad trees, Deletion, B-trees, Hashing: initial hash, collisions, separate chaining, Graphs; Implementation, Depth first search, breadth first search, topological numbering, connected components, Sorting; Insertion sort, Quicksort, Heap as priority queue; heapsort.

Textbook: Mark Weiss, Efficient C Programming, Prentice-Hall, 1995. Mark Weiss, Data Structures and Algorithms Analysis in C^{++}, Benjamin Cummings. Watts Humphrey, Introduction to Personal Software Process, Addison Wesley.
Laboratory work: Computer programming in C and C^{++}.

SC 414: Introduction to Computer Architecture *(30 hrs, no credits)*
Combinational logic networks, Computer arithmetic; arithmetic/logic unit, Sequential logic networks, Memory hierarchy, CPU design, I/O architecture, Instruction sets, addressing modes, linking and loading, Subroutines, ALU design, Basic processor design, Basic pipelining, Memory hierarchy design, Input/output, Parallel processing.

Textbook: Hennessy and Patterson, Computer Organization and Design: the Hardware/Software Interface, Morgan Kaufmann. M. M. Mano and C. R. Kime, Logic and Computer Design Fundamentals, 1997, Prentice Hall.
Laboratory work: Computer programming in assembler and electronics lab.

SC 415: Programming and electronics laboratories *(45 hrs, no credits)*
Programming laboratory: Language constructs; Variables, assignments, loops, decision structures, input/output, files, subprograms/procedures, numeric and nonnumeric data. Design and construction of software; Top down and bottom up design, decomposition, structuring, design for reuse, documentation, study of examples, writing software as a team, using software from others. Programming assignments; A variety of progressively more complex assignments.
Electronics laboratory: Review of basic features of computer hardware and software; Lab: Introduction to equipment, demo, simple experiment. Input/Output (I/O) concepts and examples; Lab: Experiment involving parallel I/O. More concepts and examples; Lab: Experiment involving serial I/O. Interfacing to the analog world; Lab: Experiment using digital-to-analog (D/A) conversion. Lab: Experiment using analog-to-digital (A/D) conversion. Techniques for analysis of acquired data; Lab: Experiment requiring digital signal processing. Interfacing to local area network (LAN); Lab: Experiment using LAN.

Textbook: Programming laboratory: C: Deitel and Deitel, C How to Program, Prentice-Hall. C++: Deitel and Deitel, C++ How to Program, Prentice-Hall. Java: Deitel and Deitel, Java How to Program, Prentice-Hall.
Electronics laboratory: None.
Laboratory work: Computer programming and electronics labs.

SC 416: Seminar *(no credits*)

SC 531: Database systems *(2 credits*)
Introduction; An overview of a database management system, The entity-relationship model, Logical organization of databases; The relational model, Relational algebra, SQL, Examples of existing relational database management systems, Physical organization of databases; Characteristics of disks and disk storage, Storage of relations, Indexing: B-trees and hashing, Query processing and optimization, Concurrency control; Transaction, Serializability, Locking, Logging and recovery, Distributed databases, Functional dependencies and normal forms, Information services for unstructured data.

Textbook: Elmasri and Navathe, Fundamentals of Database Systems, 2nd edition, Benjamin/Cummings.
Laboratory work: Computer programming using database management packages such as Informix, Oracle and FoxPro on PCs and UNIX workstations.

SC 532: Combinatorial mathematics *(2 credits)*
Introduction, Numbers and counting, Subsets, partitions, permutations, Recurrence relations and generating functions, The principle of inclusion and exclusion, Latin squares and SDRs, Extremal set theory, Steiner triple systems, Finite geometry, Ramsey's theorem, Graphs, Posets, lattices, and matroids, Automorphism groups and permutations, Enumeration under group action, Designs, Error-correcting codes.

Textbook: Peter Cameron, Combinatorics.
Laboratory work: None.

SC 533: Introduction to parallel computing *(2 credits*)
Ideal and real machine models; Vector (pipelined) processors, array machines, shared-memory multiprocessors, message-passing multiprocessors, others; programming constructs native to each class of machine. Programming models and their languages; Data-parallel models (array parallelism, parallel loops), process-based models; illustrative examples, such as matrix multiplication, sorting, and the n-body problem. Cost models and efficiency analysis of parallel programs, Parallel programming

issues; Locality, grain size, scheduling, load balancing, data distribution and alignment, communication analysis, synchronous programming, determinacy and nondeterminacy. Debugging parallel programs, Performance measurement, evaluation, and tuning, Case studies; Simple case studies from application areas such as computational fluid dynamics, computational biology, and operations research. Discrete event simulations, Grid-structured computations, Tree-structured computations, Sparse and dense linear systems, Parallelizing, Compilers.

Textbook: Almassi-Gottlieb, Highly Parallel Computing, 2nd edition, Benjamin Cummings, 1994. Michael Quinn, Parallel Computing: Theory and Practice, 2nd edition, McGraw Hill, 1994. Kumar, Grama, Gupta, and Karypis, Introduction to Parallel Computing, Benjamin Cummings, 1994. Ian Foster, Design and Building of Parallel Programs, Addison Wesley.
Laboratory work: Implementation of parallel algorithms on one or more classes of parallel computers. Emphasis is on numerical algorithms.

SC 534: Programming language design and compilers *(2 credits)*
Language Design: Elements of imperative languages, Data types: arrays, lists, user-defined types, Functional programming, Control operations, Object-oriented programming, Types Compilation: Lexical analysis: transition diagrams, regular expressions, using lex, Syntactic analysis: context-free grammars, top-down and bottom-up parsing, using yacc, Abstract syntax; syntax-directed translation, Code generation.

Textbook: Ravi Sethi, Programming Languages, 2nd edition, Addison-Wesley. Jim Holmes, Building Your Own Compiler With C++, Prentice Hall.
Laboratory work: Computer programming on workstations.

SC 535: Operating system design *(2 credits)*
Processes and concurrent programming; Basic concepts: states, transitions. Mutual exclusion, synchronization, semaphores, monitors, Ada rendezvous. Deadlock and indefinite postponement; prevention, avoidance, detection, recovery. Operating system components; Real and virtual memory; paging and segmentation; fetch, placement, and replacement algorithms; thrashing. Processor scheduling; disk space management and allocation; seek and rotational optimization; blocking and buffering. File systems; directory, structures; access methods; access control. Advanced topics; Performance evaluation. Distributed and parallel operating systems. Object orientation. Security and protection; encryption. Case Studies.

Textbook: Either H. M. Deitel, Operating Systems, Prentice Hall; or Silberschatz and Gavin, Operating System Concepts, 5th edition, Addison Wesley.
Laboratory work: Computer programming on UNIX workstations.

SC 536: Graph Theory *(2 credits)*
Introduction, Graphs: Graphs and simple graphs; Graphs isomorphism; The incidence and adjacency matrices; Vertex degrees; Paths and connection; Cycles and the shortest path problem, Trees: Trees; Cut edges and bonds; Cut vertices; Cayley's formula and Kruskal's algorithm, Connectivity: Connectivity; Blocks and construction of reliable communication networks, Euler Tours and Hamilton Cycles: Euler tours; Hamilton cycles; The Chinese postman problem and the travelling salesman problem, Planar Graphs: Planar graphs; Dual graphs and Euler's formula, Networks: Flows; Cuts; The Max-Flow Min-Cut theorem and applications.

SC 537: Computer Networks & Distributed systems *(2 credits)*
Introduction; Examples of computer networks and distributed systems, Concept of layered architecture. ISO reference model of Open System Interconnection. Overview of communication subnetworks; Physical layer protocol issues, Data link layer protocols, Network layer protocol issues:

Virtual circuits vs. datagrams. Local-area network architectures, Satellite and packet radio networks. Point-to-point packet switched networks. Models of network interconnection. Standard network access protocols. Transport and session protocol design issues; Transport connection and connection establishment, Flow control and buffering. Synchronization in distributed environment. Multiplexing. Crash recovery. Networking facilities in well-known systems. Presentation layer protocols; Terminal handling and protocols. File transfer protocol design issues. Network security and privacy. Standards for presentation layer protocols. Distributed operating system design; Models and primitives of distributed computing, Distributed resource management and scheduling: File allocation. Load sharing. Task assignment, etc. Distributed database systems; Concurrency control and synchronization. Current topics. Examples of distributed database systems.

Textbook: Sape Mullender, Distributed Systems, 2nd edition, Addison-Wesley.
Laboratory work: None.

SC 538: Artificial Intelligence (*2 credits*)
Introduction; Organization and overview, Program inspection: CHAT, a question answering program, Introduction to LISP, Program inspection: Robert's computer vision program, LISP II, Knowledge representation and deduction; Knowledge representation and valid arguments, Propositional Predicate Calculus (PPC): syntax and semantics, Deduction in PPC: rule-based systems & search spaces, Deduction in PPC: conjunctive goals & answer extraction, First-Order Predicate Calculus (FOPC): syntax and semantics, Representing knowledge in FOPC, Unification, Deduction in FOPC. Problem solving and search; Game trees and search, Basic search techniques, Problem solving: partial instantiate and prune, Generation of search heuristics. Planning; Introduction to planning. Non-linearity and protection intervals. Plan representation. Natural language processing; Introduction to natural language, Case grammar, Conceptual analysis, ATNs and review. Computer vision (understanding line drawings); Guzman's system & Huffman labeling, Waltz labeling and constraint propagation.

Textbook: Stuart Russell and Peter Norvig, Artificial Intelligence: A Modern Approach, Prentice-Hall, 1995. Patrick Winston and B. K. Horn, LISP, 3rd edition, Addison Wesley.
Laboratory work: Computer programming in LISP.

SC 539: Advanced topics in computer graphics *(2 credits)*
Review of computer graphics fundamentals, Nonparametric object representations; Conics, algebraic surfaces, bump functions. Parametric object representations; Quadrics, superquadrics, splines. NonEuclidean representations; Fractals, particle systems. Rendering; Lighting models, fast-Phong algorithm, A-buffer, V-buffer, radiosity. Ray-tracing algorithms; Distributed methods, space subdivision, parallel methods. Texture mapping, Animation; Key-frame systems, animation languages, kinetic vs. dynamic systems, modeling human and animal motion. Scientific data visualization.

Textbook: Hearn and Baker, Computer Graphics - C Version, 2nd edition, Prentice Hall.
Laboratory work: Computer programming on UNIX workstations.

SC 546: Software engineering *(2 credits*)
The software life cycle: cost of software and hardware, Software quality, User requirements, System specifications; Survey of abstract data types, axiomatic and operational techniques, concurrency. Design fundamentals; Abstract machines, stepwise refinement, simulation, bottom-up approach, modularity, extensions from a nucleus, techniques for real-time systems. Design techniques, Programming; Language requirements for software engineering, language specifications, high-level/low-level/assembler tradeoffs, concurrency, real-time programming, team programming,

optimization. Testing: theoretical and empirical; cost of testing, Verification: Partial correctness, proving termination, Maintenance; Portability, adaptability, modification, distribution. Reliability; Redundancy, error detection, fault-tolerance, faults, failures, recovery. Protection and security, Management of software projects: manpower, case studies.

Textbook: Roger Pressman, Software Engineering: A Practitioner's Approach, 4th edition,
Laboratory work: Computer programming on workstations.

SC 547: Computer architecture (*2 credits*)
Technology and performance, Instruction set architectures, Computer arithmetic, Central processing units, Pipelining, Memory hierarchies, Input-output mechanisms, Vector and multiprocessors, Parallel programming features, Case studies.

Textbook: Patterson and Hennessy, Computer Organization and Design: the Hardware/Software Interfaces, Morgan Kaufmann, 1994. Stone, High-Performance Computer Architecture, 3rd edition, Addison Wesley, 1993.
Laboratory work: None.

SC 548: Systems analysis/Systems engineering (*2 credits*)
Introduction: Systematic Thinking, Systems Definitions, Classification of Systems, Computer Based Systems.
Systems Analysis: Models of Systems, Formal Methods UML, Logical Algebra, Automata, Simulations.
System Engineering: System Life Cycle Processes, System Life Cycles Stages, CASE, Sociological Systems, Soft System Methodology, System Philosophy, Review & Trends.

SC 549: Artificial Neural Networks (*2 credits*)
Elementary neurophysiological principles, Artificial neuron models, Single layer networks (perceptions), Multi-layer feed forward networks (+back propagation), Cascade correlation (correlation training), Recurrent networks (Hopfield), Self-organizing maps (Kohonen maps), Bi-directional associative memory, Counter propagation networks, Adaptive resonance theory, Spatiotemporal sequences, Hardware realization of neural networks, Individual projects.

Textbook: Simon Haykin, Neural networks: A Comprehensive Foundation, Macmillan, 1994.
Laboratory work: Computer programming on UNIX workstations.

SC 550: Linear programming (*2 credits*)
Review, Geometry in IRn, Introduction to linear programming, Further topics, Integer programming, Special types of linear programming problems.

Textbook: Kolman and Beck, Elementary Linear Programming with Applications, Academic Press, 1980.
Laboratory work: None.

SC 551: Communication networks for computers *(2 credits)*
Overview; Examples and concepts of layered architecture; overview of higher layer protocols. Transport Layer; Internet addressing and Internet protocols; socket interface; TCP/IP protocols; client-server models. Network layer; Taxonomies; relevant parameters of network and traffic. Performance evaluation and queuing theory. Multiple-access methods for broadcast networks; Taxonomies of multiple access methods; contention methods; polling methods; reservation methods. Switched networks; Architectures of switches: circuit, packet, and ATM switches; scheduling and

admission control; routing, flow control, and congestion control. Interconnections of networks, Logical data link protocols.

Textbook: A. Tanenbaum, Computer Networks, 3rd Edition. D. Comer, Internetworking with TCP/IP, 2nd Edition, Volume 1. Fred Halsall, Data Communications, Computer Networks, and Open Systems, 3rd edition, Addison-Wesley.
Laboratory work: None.

SC 552: Digital image processing *(2 credits)*
Introduction to image processing, Elements of a digital image processing system; image acquisition, storage, processing, transmission and display. Image processing fundamentals; human vision system, sampling and quantization (spatial and brightness resolution), pixels and their relationships. Digital image processing techniques; image enhancement and restoration, pixel point processing, pixel group processing, frequency domain processing (Fourier transform), geometric transformations, image analysis, segmentation, feature extraction. Image compression and transmission, run-length encoding. Coding systems; error detection and correction, data compression schemes. Pattern recognition; basic concepts, clusters, decision functions, cluster seeking algorithms.

Textbook: Digital Image Processing, Gregory A. Baxes, SR 621.367
Other reading: Digital Image Processing; Remote Sensing and Image Interpretation, T. M. Lillesand and R. W. Kiefer; Remote Sensing Digital Image Analysis, J. A. Richards

SC 553: Project management *(2 credits)*
Principles of Project Planning, Project Initialisation, Project Life Cycles & Planning, Identifying Tasks and Estimating, Product Planning, Quality Issues, Anticipating Problems & Motivation, Financial Issues, Applying Principles.

SC 554: Special topics in Computer Science *(2 credits)*
Lecture course in topics of current interest.

Textbook: Depends on the topic.
Laboratory work: Depends on the topic.

SC 555: Laboratory work *(2 credits)*
Students of the batch are organized into teams of four to six students with an academic advisor to analyze a problem proposed, to select a suitable solution, and to implement that solution. Students work in teams to solve typical commercial or industrial problems. Work involves planning, design, and implementation. Oral and written work is required.

SC 597: Seminar *(1 credit)*
Students of the batch are organized into teams of four to six students with an academic advisor to search recent issues on a topic selected from the INTERNET and to prepare a report. Oral and written work is required.

SC 599: Research Project *(6 credits)*
Students will be required to carry out an independent research project on a topic which requires a fair amount of computer programming or computer hardware/programming. The candidates will be given the option of selecting a research problem in a preferred area that falls within the disciplines of courses undertaken. At the end of the research project the candidates are required to present their results in the form of a dissertation and a seminar.

7. PROGRAMME EVALUATION

Programme evaluation will be as stipulated in the PGIS Handbook 2002.

8. TEACHING PANEL

Dr. P. M. K. Alahakoon, Dept. Agric. Engineering, Faculty of Agriculture, Univ. Peradeniya
B.Sc. Eng. (S. Lan.), M.Sc. (VPI & SU), Ph.D. (UMC)

Prof. G. Chroust, Head, Dept. of System Engineering and Automation, Univ. Linz, Austria *Dip.Eng. (Vienna), M.Sc. (Phil.), Ph.D. (Vienna)*

Dr. F. M. R. Corea, Pricewater koopers (PVT.) Ltd., Colombo
B.Sc. Eng (Perad.), Dip. Com. Sc. (Cambridge), Ph.D. (New Castle, Upon Tyne)

Mr. V. Herath, Dept. Electrical and Electronic Engineering, Faculty of Engineering, Univ. Peradeniya *B.Sc. Eng. (Perad.), M.Sc. (Perad.)*

Prof. S. R. H. Hoole, Dept. Computer Sciences, Faculty of Engineering, Univ. Peradeniya
B.Sc.

Dr. A. Karunananda, Faculty of Natural Sciences, Open Univ. of Sri Lanka
B.Sc. (Colomb.), M. Phil. (Kel.), Ph.D. (Keele)

Dr. S. R. Kodituwakku, Dept. of Statistics & Computer Science, Faculty of Science, Univ. of Peradeniya *B.Sc. (Perad), M.Sc. (AIT), Ph.D. (MIT)*

Dr. K. M. Liyanage, Dept. Electrical and Electronic Engineering, Faculty of Engineering, Univ. Peradeniya *B.Sc. Eng. (Perad.), M.Eng., D. Eng. (Tokyo)*

Mrs. K. M. S. Liyanage, Dept. of Statistics & Computer Science, Faculty of Science, Univ. of Peradeniya *M.Eng. (Havana), Dip. (Linz.)*

Dr. V. Mutukumarasawmy, Dept. of Electrical and Electronic Engineering, Faculty of Engineering, Univ. of Peradeniya *B.Sc. Eng. (Perad.), Ph.D. (Cant.)*

Dr. H. M. Nazir, Dept. Mathematics, Faculty of Science, Univ. Peradeniya
B.Sc. (Perad.), Ph.D. (Japan)

Prof. M. Nordahl, Chalmers Univ. of Technology, Sweeden
M.Eng. (Stockholm), M.Sc. (Calif.), Ph.D. (Chalmers)

Dr. A. A. I. Perera, Dept. Mathematics, Faculty of Science, Univ. Peradeniya
B.Sc. (Perad.), M.Sc. (Oslo), Ph.D. (Melb.)

Dr. D. N. D. Ramanayake, Seylan Bank Ltd., Colombo
B.Sc. (S. Lan.), M.Sc. (AIT), Ph.D. (Washington)

Dr. N. Ratnayake, Dept. of Electrical and Electronic Engineering
B.Sc. (Perad.), Ph.D.

Dr. S. M. N. A. Senanayake, Dept. Statistics & Computer Science, Faculty of Science, Univ. of Peradeniya *M.Sc. (Havana), Ph.D. (Linz.)*

Prof. R. O. Thattil. Dept. Crop Science, Faculty of Agriculture, Univ. of Peradeniya
B.Sc. Agric. (Cey.), M.Sc. (Philippines), Ph.D. (VPI&SU)

Mr. R. Weerasekara, Dept. Electrical and Electronic Engineering, Faculty of Engineering, Univ. Peradeniya *B.Sc. Eng. (Perad.), M.Sc. (Perad.)*

Dr. J. Wijekulasooriya, Dept. Electrical and Electronic Engineering, Faculty of Engineering, Univ. Peradeniya *B.Sc. Eng. (Perad.), Ph.D.*

PROGRAMME COORDINATOR

Dr. Athula Perera
Faculty of Science
University of Peradeniya
Peradeniya