CURRICULUM HANDBOOK

# Master of Science [M.Sc.]

in

# COMPUTER SCIENCE

[CLASS OF 2025]

## AFRICAN UNIVERSITY OF SCIENCE AND TECHNOLOGY

## SUMMARY

| # | COURSE NAME | CODE | CREDIT UNITS | CATEGORY |
|---|---|---|---|---|
| 1 | Foundation in Mathematics for Science and Engineering | MTH 800 | 3 | Core |
| 2 | Discrete Structures | CSE 800 | 3 | Core |
| 3 | Advanced Statistical Methods | CSE 801 | 3 | Core |
| 4 | Data Mining | CSE 802 | 3 | Core |
| 5 | Artificial Intelligence | CSE 803 | 3 | Core |
| 6 | Big Data & Cloud Computing | CSE 804 | 3 | Core |
| 7 | Advanced Programming Languages | CSE 805 | 3 | Core |
| 8 | Data Structure & Algorithms | CSE 806 | 3 | Core |
| 9 | Database & Information Systems | CSE 807 | 3 | Core |
| 10 | Computer Networks | CSE 808 | 3 | Core |
| 11 | Software Engineering | CSE 809 | 3 | Core |
| 12 | Compiler Design | CSE 810 | 3 | Core |
| 13 | Operating Systems | CSE 811 | 3 | Core |
| 14 | Distributed & Wireless Systems | CSE 821 | 3 | Elective |
| 15 | Telecom Systems | CSE 822 | 3 | Elective |
| 16 | Digital & Computer Systems | CSE 823 | 3 | Elective |
| 17 | Computational Science | CSE 824 | 3 | Elective |
| 18 | Parallel & High Performance Computing | CSE 825 | 3 | Elective |
| 19 | Master Dissertation | CSE 899 | PASS/FAIL | Core |

**Note:**

- General grade and examination requirements, calculation of GPA and CGPA, and other academic requirements can be found in Chapter 4 of the AUST Student Handbook.
- A minimum of 39 Credit Units is required for a Master of Science degree in Computer Science.
- ALL *Core Courses* are compulsory.
- A student can take any of *Elective Courses* when it is offered. Please consult with your Head of Department on this beforehand.
- There is no credit load for the *Master Dissertation*, with a grade of PASS/FAIL awarded for the course. A grade of a PASS is needed for an award of a Master of Science degree in Computer Science.

## COURSE NAME – CSE 800: Discrete Structures

### *References*

*a. (Text book, title, author, and year)*

- Rosen K.H. (2012), Discrete Mathematics and Its Applications, McGraw-Hill Companies, 7th Edition.
- Seymour Lipschutz & Marc Lipson(2007), Discrete Mathematics (Schaum's outline), McGraw-Hill Companies.
- Cormen T.H., Leiserson C.E., Rivest R.L., & Stein C. (2009), Introduction to Algorithms, MIT press.
- JP Tremblay and R Manohar, Discrete Mathematical Structures with application to Computer Science, TMH

*b. (other supplemental materials)*

- Rafael Pass.R and Tseng W.D. A Course in Discrete tructures.

### *Specific course information*

*a. brief description of the content of the course (catalog description)*

Function, relations and sets; Basic logic; Proof techniques; Combinatorics; Trees and graphs; lattices, Boolean algebra

### *Specific goals for the course*

- Demonstrate a good understanding of mathematical reasoning
- Demonstrate a good understanding of discrete structures such as set, relations, graphs, trees etc.
- Perform combinatorial analysis in solving counting problems
- Apply graph theory to solve network problems

### *Brief list of topics to be covered*

- Logic
- Set theory
- Proof Techniques
- Functions and Relations
- Counting and Combinatorics
- Probability
- Trees and Graphs
- Boolean algebra
- Lattices

## COURSE NAME – CSE 801: Advanced Statistical Methods

### *References*

*a. (Text book, title, author, and year)*


- Moore, David S. (1991), Statistics: concepts and controversies, 3rd ed., New York: W.H. Freeman and Company
- Muenchen, Robert A. (2011), R for SAS and SPSS Users, 2nd ed., New York et al.: Springer
- Qian, Song S. (2010), Environmental an ecological statistics with R, New York: Taylor & Francis Group
- Ross, Sheldon M. (2004), Introduction to probability and statistics for engineers and scientists, 3rd ed., Amsterdam et al.: Elsevier Academic Press


*b. (other supplemental materials)*: *Nil*

### *Specific course information*

*a. brief description of the content of the course (catalog description)*

The course is designed for acquiring professional skills and knowledge in the area of statistics. The students will be enabled to independent treatment of statistical research issues. Data analysis of typical research problems will be done in R / SPSS / Python.

*b. Prerequisites: Nil*

### *Specific goals for the course (in terms of outcomes by the student)*

- Knowledge and understanding: Knowledge of the most important statistical methods for data analysis; understanding their rationale, conditions of usage and their results.
- Applying knowledge and understanding: Identification of appropriate statistical method for data analysis; independent identification and application of functions in statistical package R.
- Making judgments: Critical reviewing of own scientific work and of original publications; interpretation of statistical analyses in the context of diverse scientific fields.
- Communication skills: Ability to present results of statistical analyses correctly and intelligibly.
- Learning skills: Ability to recognize situations in which statistical analysis is necessary. Ability to judge the appropriateness of statistical methods.

### *Brief list of topics to be covered*

- Recapitulation of basic statistical concepts Descriptive statistics (measures of location and dispersion).
- Distributions, graphical representation of data, contingency tables
- Correlation and linear regression

- Hypothesis testing
- Fundamentals of modeling
- Multiple testing and the corresponding correction methods
- Graphical presentation of higher dimensional data
- 1/2 Multivariate regression, linear and polynomial analysis of variance including interaction Factor analysis

## COURSE NAME – CSE 802: Data Mining

### *References*

*a. (Text book, title, author, and year)*

- Data Mining by Charu C. Aggarwal, Springer
- Data Mining Techniques, Second Edition, Kindle Edition by AK Pujari
- Introduction to Data Mining by Vipin Kumar
- Ian H. Witten and Eibe Frank, Data Mining: Practical Machine Learning Tools and Techniques (Second Edition), Morgan Kaufmann, 2005, ISBN: 0- 12-088407-0.

*b. (Other supplemental materials): Nil*

### *Specific course information*

*a. brief description of the content of the course (catalog description)*

Data Mining studies algorithms and computational paradigms that allow computers to find patterns and regularities in databases, perform prediction and forecasting, and generally improve their performance through interaction with data. It is currently regarded as the key element of a more general process called Knowledge Discovery that deals with extracting useful knowledge from raw data. The knowledge discovery process includes data selection, cleaning, coding, using different statistical and machine learning techniques, and visualization of the generated structures. The course will cover all these issues and will illustrate the whole process by examples. Special emphasis will be give to the Machine Learning methods as they provide the real knowledge discovery tools. Important related technologies, as data warehousing and on-line analytical processing (OLAP) will be also discussed. The students will use recent Data Mining software.

*b. Prerequisites: Nil*

### *Specific goals for the course (in terms of outcomes by the student)*

- Understanding of the value of data mining in solving real-world problems.
- Understanding of foundational concepts underlying data mining.
- Understanding of algorithms commonly used in data mining tools.
- Ability to apply data mining tools to real-world problems.
- Ability to mine the data using R package

By the end of the module, the student should:

- Display a comprehensive understanding of different data mining tasks and the algorithms most appropriate for addressing them.
- Evaluate models/algorithms with respect to their accuracy.
- Demonstrate capacity to perform a self directed piece of practical work that requires the application of data mining techniques.
- Critique the results of a data mining exercise.
- Develop hypotheses based on the analysis of the results obtained and test them.

- Conceptualize a data mining solution to a practical problem.

***Brief list of topics to be covered***

- Introduction to Data Mining
- Data Warehouse and OLAP
- Data Warehouse and DBMS
- Multidimensional data model; Multidimensional Cube structures
- OLAP operations
- Data preprocessing
- Dimensionality reductions
- Data mining knowledge representation
- Attribute-oriented analysis
- Data mining algorithms: Association rules
- Data mining algorithms: Classification
- Data mining algorithms: Prediction
- Clustering
- Advanced techniques, Data Mining software and applications

## COURSE NAME – CSE 803: Artificial Intelligence

### *References*

*a. (Text book, title, author, and year)*

- *Artificial Intelligence: A Modern Approach,* Third Edition Stuart Russell and Peter Norvig, 2010. Pearson Education, Inc. ISBN: 978-0-13-604259-4

*b. (Other supplemental materials):* Several materials and handouts will be provided during the lecture

### *Specific course information*

*a. brief description of the content of the course (catalog description)*

An introduction to the basic principles, techniques, and applications of Artificial Intelligence. Coverage includes knowledge representation, logic, inference, problem solving, search algorithms, game theory, perception, learning, automated planning, reasoning under uncertainty, bio-inspired optimization and agent design. Students will be introduced to programming in AI languages like Prolog & LISP. Potential areas of further exploration include expert systems, neural networks, fuzzy logic, machine learning, robotics, natural language processing, and computer vision.

b. *Prerequisites: Nil*

### *Specific goals for the course (in terms of outcomes by the student)*

- The primary objective of this course is to introduce the basic principles, techniques, and applications of Artificial Intelligence. Emphasis will be placed on the teaching of these fundamentals, not on providing a mastery of specific software tools or programming environments. Assigned projects promote a 'hands-on' approach for understanding, as well as a challenging avenue for exploration and creativity.
- Gain a historical perspective of AI and its foundations.
- Become familiar with basic principles of AI toward problem solving, inference, perception, knowledge representation, and learning.
- Investigate applications of AI techniques in intelligent agents, expert systems, artificial neural networks and other machine learning models.
- Experience AI development tools such as an 'AI language', expert system shell, and/or data mining tool.
- Experiment with a machine learning model for simulation and analysis.
- Explore the current scope, potential, limitations, and implications of intelligent systems.

### *Brief list of topics to be covered*

a. Prologue in Artificial Intelligence

- AI Tools & Programming Languages
- Knowledge representation and inference
- Operations on Data Structures

- Advanced Tree Representations
- Basic Problem-Solving Strategies
- Best-first: A Heuristic Search Principle
- Problem Reduction and AND/OR Graphs

b. Some more search
- Heuristic search (Greedy, A*, IDA*)
- State space search
- Backtracking
- Hill climbing
- Constraint search

c. Expert Systems
- Functions of an expert system
- Main structure of an expert system
- If-then rules for representing knowledge
- Developing the shell
- Implementation
- Dealing with uncertainty

d. Game Playing
- Two-person, perfect-information games
- The minimax principle
- The alpha-beta algorithm: an efficient implementation of minimax
- Minimax-based programs: refinements and limitations

e. Other topics
- Natural language processing
- Robotics
- Genetic Algorithms
- Neural Networks
- Machine Learning

## COURSE NAME – CSE 804: Big data and Cloud Computing

### *References*

*a. (Text book, title, author, and year)*

- Code Complete, 2nd edition, by Steve McConnell, Microsoft Press, 2004.

*b. (Other supplemental materials):* Several materials and handouts will be provided during the lecture.

### *Specific course information*

*a. brief description of the content of the course (catalog description)*

In recent years, due to advancement of internet technologies and instrumentation of every part of our life, we have noticed a huge surge in data available to us. This revolution is termed as Big Data. This Big Data cannot be processed or managed by any traditional methods of processing. This has led to development of several high performance and distributed computing platforms and programming frameworks. The design of such platforms relies on distributed computing concepts which are implemented in the form of systems such as Clusters and Clouds, and Big Data frameworks such as MapReduce and Stream Computing. These systems play an important role in today's' research, academia or industries by providing the processing of data generated from a variety of networked resources, e.g. large data stores and information repositories, expensive instruments, social media, sensors networks, and multimedia services for a wide range of applications.

*b. Prerequisites: Nil*

### *Specific goals for the course (in terms of outcomes by the student)*

- The aim of this unit is to provide students with the foundation knowledge and understanding of Big Data and distributed computing systems and applications especially in context of Cloud. In other words, this unit will equip students with essential knowledge that is needed for building next-generation applications that are scalable and efficient and can process Big Data.
- The unit will also explain how the business models of enterprises are changing with these forms of computing that provide large storage and computation space without purchasing expensive computer systems.

### *Brief list of topics to be covered*

Big Data:

- MapReduce platforms,
- Stream Computing platforms and Algorithms.

Cloud computing:

- Cloud technologies,
- Resource management and scheduling,
- Application building for managing and analyzing data

## COURSE NAME – CSE 805: Advanced Programming Languages

### *References*

*a. (Text book, title, author, and year)*

- Concepts in Programming Languages – John C Mitchell, Cambridge Press, ISBN: 0521780985 (Req)
- Concrete Abstractions – An Introduction to Computer Science Using Scheme – Max Hailperin – (free supplementary e-book posted on Moodle)

*b. (other supplemental materials): Nil*

### *Specific course information*

*a. brief description of the content of the course (catalog description)*

This will be a study of the key concepts and principles of programming languages in general starting with the syntax, semantics and pragmatics of a diverse set of high-level programming languages and paradigms. The languages chosen are compared and contrasted in order to demonstrate general principles of programming language design evolution. The course emphasizes the concepts underpinning modern languages, the major paradigms of programming and driving forces behind their designs and evolutions rather than the mastery of particular language details. Programming projects will be provided in different Programming languages; at least one each of the major programming paradigms – imperative/object-oriented, functional and declarative.

*b. Prerequisites: Nil*

### *Specific goals for the course (in terms of outcomes by the student)*

Upon successful completion of this course, students should be able to:

- Explain some of the theoretical issues of computation, recognize non-computable problems and give formal arguments to support non-computability claims.
- Explain and use basic and advanced programming language concepts such as evaluation, declarations, expressions, values, data types, pattern matching, polymorphism, higher-order functions, continuations, exceptions and modularization.
- Master the foundation concepts of the lambda-calculus; its application in functional programming language like scheme and across programming languages.
- Design recursive algorithms and develop recursive programs.
- Explain the main mechanisms of language processing including interpretation, compilation, incremental compilers and virtual machines.
- Describe, compare, appreciate, and apply such language concepts as syntax, semantics, names, scopes, bindings and language mechanisms such control flow, data types and run-time execution of programs.

- Determine appropriate languages/paradigms for given applications and effectively employ them.
- Gain amble practice with a multi-paradigm language like Java, LISP, Python.

### *Brief list of topics to be covered*

- Overview of programming languages: Procedural, object oriented, functional and logic programming languages.
- Procedure oriented: Basic building blocks of languages, parameter passing mechanisms, array & records.
- Object oriented: OOP concepts, differences with procedural language, Inheritance, modeling in OOP, Modeling paradigm.
- Functional Programming: Lambda Calculus, ML, Implementation of λ-calculus.
- Logic programming: Inference rules, PROLOG

## COURSE NAME – CSE 806: Data Structures and Algorithms

### *References*

a. *(Text book, title, author, and year)*

- Jörg Arndt Algorithms for Programmers, http://www.jjj.de/fxt/#fxtbook,
- Peter Brass Advanced Data Structures, Cambridge University Press, 2008
- Thomas L. Cormen, Charles E. Leiserson, Ronald L. Rivest Introduction to Algorithms, The MIT Press, 2000.
- Adam Drozdek Data Structures and Algorithms in C++, Brooks/Cole, 2001.
- Bruno R. Preiss Data Structures and Algorithms with Object-Oriented Design Patterns in Java. Wiley and Sons, 1998

*b. (Other supplemental materials):* Several materials and handouts will be provided during the lecture

### *Specific course information*

*a. brief description of the content of the course (catalog description)*

Data management system is the backbone of all socio-economic structure (industries, services, governments, etc.) for operational activities, for understanding the evolution of indicators to assist in strategic planning. Most of the data are structured and when they are not, they are processed to obtain structured data. This topic has received a great focus for the past decades with the approached well formalized. Even with the emergence of the concept of Big Data, most of the socio-economic organizations still depend on structures data. This course focuses on programming techniques, as it relates to data structures in programming.

*b. Prerequisites: Nil*

### *Specific goals for the course (in terms of outcomes by the student)*

- This course introduces students to new types of data structures such as trees (including binary and multiway trees), heaps, stacks and queues. Students will also learn how to design new algorithms for each new data structure studied, create and perform simple operations on graph data structures, describe and implement common algorithms for working with advanced data structures and recognize which data structure is the best to use to solve a particular problem.

### *Brief list of topics to be covered*

- Complexity classes,
- Complexity of algorithms,
- Linear structures,
- Tree structures,
- Divide and conquer methods,

- Master theorem
- Greedy methods
- Python implementations of the algorithms

## COURSE NAME – CSE 807: Database and Information Systems

### *References*

a. *(Text book, title, author, and year)*

- An Introduction to Relational Database Theory, (http://bookboon.com/en/an-introduction-to-relational-database-theory-ebook)

*b. (Other supplemental materials):* Several materials and handouts will be provided during the lecture

### *Specific course information*

*a. brief description of the content of the course (catalog description)*

This relates to building information systems based on databases. This course focuses mostly on data modeling to be developed using database management systems.

*b. Prerequisites: Nil*

### *Specific goals for the course (in terms of outcomes by the student)*

- At the end of the course, the students are expected to really understand the importance of data modeling. The course will integrate a practical application of modeling techniques.

### *Brief list of topics to be covered*

- Why data model?
- Entity-relational model
- Relational model
- Normal forms
- Deriving Relational model from Entity-relational model
- The basic components of an information system (database management system, user-interface)
- Study examples of modeling relational model tools
- Study the use of a database management tool
- Developing an information system using a content-management system such as Drupal

## COURSE NAME – CSE 808: Computer Networks

### *References*

*a. (Text book, title, author, and year)*

- Computer Networks, Fifth Edition: A Systems Approach, Larry Peterson Bruce Davie (The Morgan Kaufmann Series in Networking) 5th Edition, 2011.

*b. (Other supplemental materials): Nil*

### *Specific course information*

*a. brief description of the content of the course (catalog description)*

Computer Networks. Network Layers. Performance. Encoding, Error Detection and Correction. Reliable Transmission. Ethernet & Internetworking. Routing. Transport Layer. TCP/IP. TCP Window & Congestion. Network Security. Graph Theory.

*b. Prerequisites: Nil*

### *Specific goals for the course (in terms of outcomes by the student)*

- The course presents the fundamental principles of computer networks and data communications. Emphasis is given on current technologies and architectures for establishing direct link and packet-switched networks, sharing access to a common communication medium, inter-networking and routing, end-to-end flow control, congestion control and recourse allocation, and network security.

### *Brief list of topics to be covered*

- Introduction to Networks and Networking
- Physical Layer: Channel capacity, Nyquist condition
- Data Link Layer: parity, coding
- Data Link Layer: code construction
- MAC Sublayer: Stop and wait, sliding window
- MAC Sublayer: CSMA/802.11
- Markov chains and queueing: M/M/1 and M/M/c/c
- Network Layer: Routing Algorithms
- Network Layer: IP
- Transport Layer: Flow control
- Transport Layer: Congestion Control
- Network Security
- Graph theory and general networks
- Branching processes and random graphs

## COURSE NAME – CSE 809: Software Engineering

### *References*

a. *(Text book, title, author, and year)*

- "Software Engineering.", Sommerville, 9*th Edition, Addison Wesley, 2010.*
- "Applying UML and Patterns.", Craig Larman,2*nd Edition, Prentice Hall, 2002*

*b. (Other supplemental materials): Nil*

### *Specific course information*

*a. brief description of the content of the course (catalog description)*

This course is aimed at helping students build up an understanding of how to develop a software system from scratch by guiding them thru the development process and giving them the fundamental principles of system development with object oriented technology using UML. The course will initiate students to the different software process models, project management, software requirements engineering process, systems analysis and design as a problem-solving activity, key elements of analysis and design, and the place of the analysis and design phases within the system development life cycle.

*b. Prerequisites: Nil*

### *Specific goals for the course (in terms of outcomes by the student)*

- Develop an understanding of project management, software process models and the ability to select the suitable model to use in software development.
- Develop an understanding of requirements engineering process and distinguish between different types of requirements.
- Ability to analyze, design and develop the system models using object oriented methodology (UML) for software development.
- Ability to prepare the software requirements specification document for a software project.
- Demonstrate the ability to research a particular topic and develop it for a specific audience and purpose.
- Develop and empower the presentation skills.
- Develop the teamwork management skills.

### *Brief list of topics to be covered*

- Introduction to Software Engineering.
- Software Processes.
- Project Management.
- Requirements Engineering.
- Structured Analysis and Design.

- System Modeling: Introduction to OO Analysis and Design (UML), Use Cases, Sequence Diagrams, Conceptual Modelling, Class Diagrams

## COURSE NAME – CSE 810: Compiler Design

### *References*

*a. (Text book, title, author, and year)*

- Compilers – Principles, Techniques and Tools by Alfred V. Aho, Monica S. Lam, Ravi, Sethi, and Jeffrey D. Ullman. Addison-Wesley. Second Edition, 2007, Available for FREE at https://drive.google.com/file/d/0B1MogsyNAsj9elVzQWR5NWVTSVE/view
- Modern Compiler Implementation in Java by A.W. Appel and J. Palsberg. Cambridge University Press 2002, Available for FREE at:https://eden.dei.uc.pt/~amilcar/pdf/CompilerInJava.pdf
- Compiler Design: Theory, Tools, and Examples by Seth D. Bergmann. WCB, 2016, Available for FREE at:http://elvis.rowan.edu/~bergmann/books/cd/java/CompilerDesignBook.pdf

*b. (Other supplemental materials):* Several materials and handouts will be provided during the lecture

### *Specific course information*

*a. brief description of the content of the course (catalog description)*

A compiler is a computer program (or a set of programs) that transforms source code written in a programming language (the source language) into another computer language (the target language), with the latter often having a binary form known as object code. A compiler is likely to perform many or all of the following operations: lexical analysis, preprocessing, parsing, semantic analysis (syntax-directed translation), code generation, and code optimization. Program faults caused by incorrect compiler behavior can be very difficult to track down and work around; therefore, compiler implementers invest significant effort to ensure compiler correctness. Compilers are fundamental and vital subject in computer science. It is a subject that has been studied intensively since the early 1950's and continues to be an important research field today. Compilers are important part of the computer science curriculum for many reasons:

- It provides students with a better understanding of and appreciation for programming languages.
- The techniques used in compilers can be used in other applications with command languages.
- It provides motivation for the study of theoretic topics.*b. prerequisites*

*b. prerequisites:* The following topics are helpful to understand compilers:

- Automata theory
- Programming languages
- C/Java programming
- Data Structure and Algorithms, etc.

### *Specific goals for the course (in terms of outcomes by the student)*

- Understand the role of languages processing systems such as compilers and translators,

- The processing methods of languages,
- The importance of compilers,
- The relation between theory and practice (that is, formal language theory and languages processing systems),
- How the compilers' components work, and
- How to design and implement the compiler modules.

***Brief list of topics to be covered***

- An overview: Anatomy of the compiler. The compiler components and the compiler's role in the programming language processing. Review of automata and formal languages and its use in compiler.
- Mathematical background: The necessary mathematical theories will be introduced such as sets, graphs, tress, proof techniques, etc.
- Automata Theory: Finite sate machines and regular expressions are necessary to understand the compiler front-end construction and how it works.
- Lexical Analysis (Scanner): The role of the scanner in the compiler. Lexical tokens. Regular expressions. Specifying lexical structures using regular expressions. Using finite automata to recognize lexical structures. Conversion between finite automata and regular expressions.
- Syntax Analysis (Parser): The parser's role in the compiler. Top-down parsing techniques.
- LL(k) parsing. Bottom-up parsing techniques. LR(k) parsing. Syntax-directed translation.
- Automatic compiler generators: Scanners and parsers can be generated automatically. Tools that can be used in such automatic generation will be covered.
- Semantics analysis: Introduction to type checking and type system. Type equivalence. Static and Dynamic checks.
- Intermediate representations: Intermediate languages. Abstract Syntax Trees. Directed Acyclic Graphs (DAGs). Control Flow Graphs. Stack based (Postfix) representations.
- Code optimization: Short introduction for optimization techniques will be covered.
- Code generation: Code generation from intermediate code.
- Issues in the design of a code generator will be discussed.

## COURSE NAME – CSE 811: Operating Systems

### *References*

*a. (Text book, title, author, and year)*

- Operating System Concepts 8th Edition, by Abraham Silberschatz. Peter B. Galvin , Greg Gagne. 2009

*b. (Other supplemental materials): Nil*

### *Specific course information*

*a. brief description of the content of the course (catalog description)*

Operating System Structures. Processess. Threads. Scheduling. Synchronization. Deadlocks. Memory Management. File System Interface & Implementations. Mass Storage. File Protection. Input/Output

*b. prerequisites: Nil*

### *Specific goals for the course (in terms of outcomes by the student)*

- This course provides a graduate-level introduction to the theory and design of multi-programmed operating systems. Some of the topic areas covered include concurrent processes, process communication, input/output supervisors, memory management, resource allocation, and process scheduling. Selected topics in distributed operating systems will also be addressed.

### *Brief list of topics to be covered*

- Operating System Overview
- Introduction
- OS Structures
- Processes
- Threads
- Scheduling
- Synchronization
- Deadlocks
- Memory Management
- File System
- Case studies of various OS

## COURSE NAME – CSE 821: Distributed and Wireless Systems

### *References*

*a. (Text book, title, author, and year)*

- Distributed Operating Systems & Algorithms, Randy Chowand Theodore Johnson, Addison Wesley, 1997.
- Distributed Systems, Concepts and Design, Georges Coulouris, Jean Dollimore and Tim Kindberg, Addison-Wesley, 2001.
- Distributed Systems, Principles and Paradigms, Andrew S. Tanenbaum and Maarten van Steen, Prentice Hall, 2002.
- Concurrent Systems, 2nd edition, Jean Bacon, Addison Wesley, 1997.
- Time, clocks and the ordering of events in a distributed system, L. Lamport. Comm. ACM, vol.21,7, July 1978.
- Ad hoc networking, Perkins and C. E. Addison-Wesley Verlag, 2001.
- Wireless sensor networks, I. Akyildiz. John Wiley & Sons, Inc., 2010.

*b. (Other supplemental materials)*

- Distributed systems: towards a formal approach, G. LeLann. IFIP Congress, Toronto, August 1977.
- Ad hoc networking, Perkins and C. E. Addison-Wesley Verlag, 2001.
- Wireless sensor networks, I. Akyildiz. John Wiley & Sons, Inc., 2010.

### *Specific course information*

*a. brief description of the content of the course (catalog description)*

Position of distributed systems among the various existing computer architectures. Understand the context of the appearance of distributed systems. Define distributed systems and their objectives. Distinguish between distributed systems and distributed applications. Understand the implementation of distributed applications. Understand operation principles of the wireless communication systems.

*b. prerequisites: Nil*

### *Specific goals for the course (in terms of outcomes by the student)*

- *The student will be able to explain the significance of current research about a particular topic.*
- The student will demonstrate the ability to explain general properties, challenges, and characteristics of distributed systems
- The student will demonstrate the ability to explain the notions of causality and time in light of the design and implementation of distributed systems
- The student will demonstrate the ability to explain general distributed algorithms for synchronization and concurrency, coordination, transactions, and replication

- The student will demonstrate the ability to compare replication schemes with respect to performance, availability, and consistency concerns
- The student will demonstrate the ability to explain practical issues that need to be considered when designing, implementing, and debugging distributed systems
- The student will demonstrate the ability to design, implement, and debug distributed systems
- The student will demonstrate the ability to employ fundamental distributed algorithms to solve problems that arise when engineering distributed systems
- The student will demonstrate the ability to explain the inner mechanisms of current production distributed systems

***Brief list of topics to be covered***

- Introduction to distributed systems
- Time, clocks and ordering of events
- Global State
- Termination Detection
- Distributed mutual exclusion algorithms
- Deadlocks
- Fault tolerance
- Election algorithms and Distributed processing
- Group communication
- Scheduling in distributed systems
- Introduction to Wireless systems
- Introduction to Ad Hoc networks

## COURSE NAME – CSE 822: Telecom Systems

### *References*

*a. (Text book, title, author, and year)*

*b. (Other supplemental materials):* Several materials and handouts will be provided during the lecture

### *Specific course information*

*a. brief description of the content of the course (catalog description)*

*b. Prerequisites: Nil*

### *Specific goals for the course (in terms of outcomes by the student)*

- The objective of the course is to impart theoretical and practical inputs to students in order to mold them into sound wireless telecom professionals.

### *Brief list of topics to be covered*

Telecom and Wireless Basics

- Telecom Basics
- Radio Wave Propagation
- Antenna Theory
- Mobile Number Portability

GSM

- Introduction to GSM
- Features of GSM
- Network Components
- Terrestrial Interfaces
- Channel Concept
- Burst formats, Coding and Interleaving
- Power Control, DTX and DRX
- Idle Mode
- Dedicated Mode

CDMA

- Introduction to CDMA
- The Cellular System
- CDMA Channels
- Features of CDMA

- Traffic Cases

WCDMA

- WCDMA Introduction
- Network Elements
- Radio Resource Management
- Channel Concept
- Protocols
- Traffic Cases
- HSDPA concept

TD-LTE

- Introduction to LTE and its unique technologies
- LTE Radio Interface Architecture
- LTE Access Procedure

Applications

- Introduction to RF Planning
- Transmission Planning
- Brief Introduction of RF Survey
- BTS Installation and Commissioning
- Microwave Installation and Commissioning
- EMF Measurements
- RF Drive Test and Optimization

## COURSE NAME – CSE 823: Digital and Computer Systems

### *References*

a. *(Text book, title, author, and year)*

- Book: Introduction to computing systems: from bits & gates to C & beyond Second Edition ISBN: 0-007-246750-9.
- Book: Computer Organization and Design, 3rd Third Edition, ISBN-13: 978-8181475343

*b. (Other supplemental materials)*

### *Specific course information*

*a. brief description of the content of the course (catalog description)*

The first part of this course covers the basics of Boolean algebra that forms the theoretical foundation on which digital circuits are built and how to aggregate circuits into larger components to create complex designs. The second part is about computer systems, their architecture, design and how they work. It encompasses the definition of the machine's instruction set architecture, its use in creating a program, and its implementation in hardware.

*b. prerequisites*

- Introduction to Computer System,

### *Specific goals for the course (in terms of outcomes by the student)*

- The course addresses the bridge between gate logic and executable software, and includes programming both in assembly language and VHDL.

### *Brief list of topics to be covered*

- Number Systems and Number Codes, Arithmetic and Logic Units, Design at the Register Transfer Level
- Machine's instruction set architecture (ISA) including basic instruction fetch and execute cycles, instruction formats, control ow, and operand addressing modes.
- Hardware description language, HDL (e.g., either VHDL or verilog) including their uses, structural, and behavioral descriptions
- Design and functioning of a machines central processing unit (CPU) including the datapath components (ALU, register level) and the control unit.
- Basic input/output functioning including program controlled I/O and interrupt I/O.
- Organization of memory hierarchies including the basics of cache design and DRAM architectures.
- Multicore and Multiprocessors
- Performance of processors and caches

## COURSE NAME – CSE 824: Computational Science

### *References*

*a. (Text book, title, author, and year)*

- Texts in Computational Science and Engineering 6 – A Primer on Scientific Programming with Python, Second edition, Hans Petter Langtangen, 2011
- Texts in Computational Science and Engineering 3 – Python Scripting for Computational Science, Third edition, Hans Petter Langtangen, 2008.
- Advanced Engineering Mathematics, Tenth Edition, Erwin Kreyszig, Herbert Kreyszig, Edward J. Norminton, John Wiley & Sons, Inc., 2011.

*b. (other supplemental materials)*

- Udemy, Python Programming Language 3.4.2 with GUI Tutorial 2011.
- Python Programming for Beginners, A Step-by-Step Guide to Learning the Basics of Computer Programming and Python Computer Language, Corey Kidd, 2015.
- Texts in Computational Science and Engineering 2 – Scientific Computing with MATLAB and Octave, Fourth Edition, Alfio Quarteroni, Fausto Saleri, Paola Gervasio 2014.

### *Specific course information*

*a. brief description of the content of the course (catalog description)*

Introduction to a scripting language (ex. Python, PhP, etc.), Solving scalar nonlinear equations with a single variable, Numerical linear algebra and linear systems of equations, Interpolation and data-fitting, Numerical differentiation and integration, Ordinary differential equations, Principles of finite precision computation, Case studies – programmed case studies.

*b. Prerequisites or co-requisites*

- Calculus II
- Linear algebra for engineers
- Linear algebra

### *Specific goals for the course (in terms of outcomes by the student)*

- The student will demonstrate the ability to install, launch and run Python
- The students will demonstrate understanding of the difference between Compiler and Interpreter
- The students will demonstrate understanding of Script Programming using Python
- The students will demonstrate understanding of developing Modules in Python
- The students will demonstrate understanding of Nonlinear equations
- The students will demonstrate understanding of Bisection method
- The students will demonstrate understanding of Newton's method
- The students will demonstrate understanding of linear algebra
- The students will demonstrate understanding of LU decomposition with pivoting, norms and conditioning of linear systems

- The students will demonstrate understanding of Sparse matrices and complexity of numerical algorithms
- The students will demonstrate understanding of Polynomial interpolation
- The students will demonstrate understanding of Interpolation and approximation of functions
- The students will demonstrate understanding of Least-squares data approximation
- The students will demonstrate understanding of Numerical differentiation
- The students will demonstrate understanding of Numerical integration
- The students will demonstrate understanding of Initial value problems
- The students will demonstrate understanding of Numerical solution of initial value problems in practice
- The students will demonstrate understanding of Floating-point number systems
- The students will demonstrate understanding of Errors in computation
- The students will demonstrate ability to Program all the concepts above in Python
- The students will develop Programming modules for the concepts above in Python

***Brief list of topics to be covered***

- Introduction
- Basic Python
- More Python programming
- Few last notes on Python programming
- Nonlinear equations
- Bisection method
- Newton's method
- Introduction to linear algebra
- LU decomposition with pivoting, norms and conditioning of linear systems
- Sparse matrices and complexity of numerical algorithms
- Polynomial interpolation
- Interpolation and approximation of functions
- Least-squares data approximation

## COURSE NAME – CSE 825: Parallel and High Performance Computing

### *References*

*a. (Text book, title, author, and year)*

- Multicore and GPU Programming: An Integrated Approach, Gerassimos Barlas, Morgan, Kaufman/Elsevier, 2015.
- Pro Git, Scott Chacon and Ben Straub, Apress, 2014. https://git-scm.com/book/en/v2
- Introduction to High-Performance Scientific Computing, Victor Eijkhout, 2015. https://bitbucket.org/VictorEijkhout/hpc-book-and-course/downloads/EijkhoutIntroToHPC.pdf
- Parallel Computing for Science and Engineering, Victor Eijkhout, 2015. https://bitbucket.org/VictorEijkhout/parallel-computing-ook/downloads/EijkhoutParComp.pdf
- Designing and Building Parallel Programs, Ian Foster, 1995. http://www.mcs.anl.gov/~itf/dbpp/text/book.html
- High Performance Computing, Charles Severance, 1998. http://cnx.org/content/col11136/latest/
- MPI: The Complete Reference, Marc Snir, Steve Otto, Steven Huss-Lederman, David Walker, and Jack Dongarra, 1996. http://www.netlib.org/utk/papers/mpi-book/mpi-book.html
- Lecture Notes from Joint SISSA/ICTP Masters in HPC for Science and Technology http://mhpc.it
- Slides from various ICTP activities on HPC and related topics including the 2016 "Introductory School on Parallel Programming and Parallel Architecture for High-Performance Computing | (smr 2877)". http://indico.ictp.it/event/7659/other-view?view=ictptimetable

*b. (Other supplemental materials):* Several materials and handouts will be provided during the lecture

### *Specific course information*

*a. brief description of the content of the course (catalog description)*

From the early 1970's, Computers and computing has been used for solving problems and challenges first in academic institutions and later on in other sectors such as government and business. Tackling large scale problems required a correspondingly large-scale computing (Supercomputers), which in the early days were built using specialized designs and hardware and were quite expensive to acquire and maintain. The introduction of Beowulf "cluster" computing (by Prof. Thomas Sterling at Indiana University) revolutionized the landscape by exploiting the parallel use of multiple normal (commodity) computers running open-source software for low-cost supercomputing. Today, thanks to mainly physical limitations, the core concepts from Beowulf

clusters have resulted in multi-core processors, the powerhouse of all computing systems from mobile devices to clusters of computers (supercomputers). In the last 10 years, Graphics Processing Units (GPU) hardware devices have also been used for general numeric computation, thanks to their higher efficiency in executing Single Instructions on Multiple Data (SIMD) simultaneously. Now-a-days, it is possible to purchase specialized General Purpose Graphics Processing Unit (GPGPU) and Many Integrated Core (MIC) add-on cards for workstations (computers) that allow one to boost or scale the performance of a single desktop computer for supercomputing. Indeed, maximizing the performance of computing systems that are built with multi-core processors requires parallel programming which often implies a change in thinking and software development. Although, modern programming languages and compilers can to certain extents exploit multicore processing automatically, careful, thoughtful, and creative programing is still required in many instances. For example, developing parallel (or multicore) versions of some algorithms may be easy while others may require clever restructuring to attain even a moderate level of improvements over equivalent serial computing versions. This course on High Performance Computing (HPC) builds on the concepts and management of multiprocessing and multi-threaded programming as they are used in cluster computing to address problems requiring large amounts of computational power, memory, or both.

*b. prerequisites:* A brief introduction to programming in C will be provided in the course. Prior exposure to serial and parallel programming will be helpful, as well as experience with linear algebra/matrices.

### *Specific goals for the course (in terms of outcomes by the student)*

- The aim of this unit is to provide students with the foundation knowledge and understanding of Big Data and distributed computing systems and applications especially in context of Cloud. In other words, this unit will equip students with essential knowledge that is needed for building next-generation applications that are scalable and efficient and can process Big Data.

- The unit will also explain how the business models of enterprises are changing with these forms of computing that provide large storage and computation space without purchasing expensive computer systems.

On completing this course, learners will be able to design and implement parallel programs on multi-core, cluster, and GPGPU architectures. More specifically they will be able to:

- analyze a programming task and identify what portions admit a parallel implementation
- use OpenMP to develop applications for multi-core computers
- use the MPI standard to develop applications for clusters
- use CUDA and Thrust to develop applications for GPGPU hardware
- understand and address issues arising in installation and management of HPC clusters

### *Brief list of topics to be covered*

- Introduction and History: What is parallel and high performance computing?

- Tools for parallel and high performance computing (Scripting, numerical libraries, large file support via HDF5, OpenMP, MPI, CUDA/OpenCL, etc.)
- High performance issues (memory hierarchy and caching, bandwidth, data I/O)
- Parallel computation issues (partitioning, synchronization, load balancing)
- Survey of problems (differential equations, linear algebra, sorting, searching)