CSE Course Syllabi

Module Code	CS 1952	Module Title	Engineering Design			
Credits	1.5	Hours/Week	Lectures	2	Pre – requisites	None
GPA/NGPA	NGPA		Lab/Assignments	2/1		

Learning Outcomes

After completing this module, students should be able to

- explain basic engineering design process
- demonstrate the ability to solve a problem in the engineering domain using engineering design process

Outline Syllabus

- **Design Principles**
  - Introduction to Engineering Design
  - Life Cycles of Engineering products and processes
  - Design process and Design Tools
  - Concurrent Engineering, Creativity and Reasoning
  - Analysis, synthesis, simulation, evaluation and Decision Making.
- **Case Studies**
  Several simple but comprehensive design case studies selected from different disciplines of engineering addressing the topics,
  - Design for Manufacturing
  - Software & Hardware aspects in design
  - Mechanical and material aspects in design
  - Electrical, Electronic and IT aspects in design
- **Design Assignments**
  Group based design assignments (Topics to be selected by Engineering Design Centre in consultation with the Department)
  The design assignment is formulated to develop the application of design concepts to engineering problems. Through a simple design, the students are expected to develop teamwork, innovation, choice of materials, survey skills, pricing and marketing skills and manufacture.
  The project will include:
  - gathering of data and infom1ation from various sources as a preliminary to the design
  - preparing a work plan and delegating duties
  - working with others and to produce results by given deadlines and within given costs
  - learning the basic procedures required for conceptual, preliminary and detailed designs
  - learning the importance or the cost component in the manufacturing process
  - preparing a report and making a presentations on the work done
  - demonstrating the working of the prototype

Module Code	**CS 1962**	**Module Title**	**Engineering Skill Development**			
Credits	**1.5**	**Hours/Week**	**Lectures**	**1**	**Pre – requisites**	**None**
GPA/NGPA	**NGPA**		**Lab/Assignments**	**1/4**		

Learning Outcomes

After completing this module, students should be able to

- demonstrate the understanding on technologies commonly used in the disciplines of engineering
- demonstrate the ability to use technologies specific to computer science and engineering

Outline Syllabus

The course is designed to develop the practical (hands on) skills of students in engineering. This is done through

- Teaching of basic tools (workshop, drawing, AUTOCAD, PSpice etc)
- Training on some tools and technologies specifically related to computer science such as web development technologies, programming languages and the installation and configuration of popular operating systems.
- Construction of a given product utilising the above experience

<table>
<tr><td>Module Code</td><td>CS 2012</td><td>Module Title</td><td colspan="4">Principles of Object Oriented Programming</td></tr>
<tr><td>Credits</td><td>3</td><td rowspan="2">Hours/Week</td><td>Lectures</td><td>2</td><td rowspan="2">Pre – requisites</td><td rowspan="2">CS1032</td></tr>
<tr><td>GPA/NGPA</td><td>GPA</td><td>Lab/Assignments</td><td>3</td></tr>
<tr><td colspan="7"><u>Learning Outcomes</u>

After completing this module, students should be able to
• analyse a system and decompose it into components based on object-oriented concepts
• implement a collection of objects and compose a system that function according to object-oriented program design
• analyse an object-oriented system and extend or modify functionality while preserving maintainability and correctness</td></tr>
<tr><td colspan="7"><u>Outline Syllabus</u>

• Philosophy of object-orientation
o Abstraction
o Encapsulation and information-hiding
o Separation of behaviour and implementation
o Division of a system into manageable components
o Reuse and simple interfaces
• Object-oriented programming
o Definition and use of classes
o Subclasses, inheritance and class hierarchies
o Polymorphism
• Object-oriented application development
o Java programming language and API
o Instance creation and scope control
o Data structures and iterable collections
o Stream based I/O and object serialization
o Event handling for interactive programs
o Concurrent processes and threads</td></tr>
</table>

<table>
<tr><td>Module Code</td><td>CS 2022</td><td>Module Title</td><td colspan="4">Data Structures and Algorithms</td></tr>
<tr><td>Credits</td><td>2.5</td><td rowspan="2">Hours/Week</td><td>Lectures</td><td>2</td><td rowspan="2">Pre – requisites</td><td rowspan="2">CS1032</td></tr>
<tr><td>GPA/NGPA</td><td>GPA</td><td>Lab/Assignments</td><td>3/2</td></tr>
<tr><td colspan="7"><u>Learning Outcomes</u>

After completing this module, students should be able to
• analyse the complexity of basic algorithms
• implement and use common data structures
• select appropriate data structures and algorithms for a given situation
• apply basic algorithm design techniques for a given situation</td></tr>
<tr><td colspan="7"><u>Outline Syllabus</u>

• Complexity analysis of algorithms
• Recursion
• Searching
• Sorting
• Basic algorithm design techniques
o Divide-and-conquer
o Greedy approach
o Dynamic Programming
• Basic data structures and operations on them
o Arrays
o Linked lists
o Queues
o Stacks
o Sets
o Trees
o Hash tables
o Graphs
• Introduction to NP-Completeness</td></tr>
</table>

<table>
<tr><td>Module Code</td><td>CS 2052</td><td>Module Title</td><td colspan="4">Computer Architecture</td></tr>
<tr><td>Credits</td><td>3</td><td rowspan="2">Hours/Week</td><td>Lectures</td><td>2</td><td rowspan="2">Pre – requisites</td><td rowspan="2">None</td></tr>
<tr><td>GPA/NGPA</td><td>GPA</td><td>Lab/Assignments</td><td>3</td></tr>
<tr><td colspan="7"><u>Learning Outcomes</u>

After completing this module, students should be able to
• list down the components of a computer and their functionalities
• describe how a program is executed in a computer
• sketch designs of controllers for various components in a computer system
• evaluate the performance enhancement mechanisms in computer systems</td></tr>
<tr><td colspan="7"><u>Outline Syllabus</u>

• General System Architecture
• Instruction Set Architecture
• Processors and Architectures
• Memory hierarchy
• Storage system and I/O Techniques
• Performance and Benchmarking</td></tr>
</table>

Module Code	MA 1032	Module Title	Numerical Methods for Computer Science			
Credits	3	Hours/Week	Lectures	3	Pre – requisites	MA 1012
GPA/NGPA	GPA		Lab/Assignments	0		

Learning Outcomes

After completing this module, the students should be able to

- understand the basic concepts of numerical methods including error analysis , methods for solving non – linear equations, methods for solving liner systems, approximations of functions, derivatives and integrals
- practically apply these methods in engineering problems

Outline Syllabus

- **Modeling, Computers, and Error Analysis**
  - Mathematical Modeling and Engineering Problem Solving
  - Programming and Software
  - Approximations and Round-Off Errors
  - Truncation Errors and the Taylor Series
- **Roots of Equations**
  - Bracketing Methods (bisection, false-position, incremental searches etc.)
  - Open Methods (fixed- point iteration, Newton-Raphson, Secant method, etc.)
  - Roots of Polynomials
  - Case Studies: Roots of Equations
- **Liner Algebraic Equations**
  - Gauss Elimination
  - LU Decomposition and Matrix Invention
  - Special Matrices and Gauss-Seidel
  - Case Studies: Liner Algebraic Equations
- **Optimization**
  - One-Dimensional Unconstrained Optimization (golden-section search, quadratic interpolation, Newton's method etc.
  - Multidimensional Unconstrained Optimization (direct and gradient methods, etc.)
  - Constrained Optimization (liner programming, non-linear constrained optimization etc.)
  - Case Studies: Optimization
- **Curve Fitting**
  - Least- Squares Regression (linear, non-linear, polynomial, multiple-linear regression etc.)
  - Interpolation (Newton's divided difference, Lagrange polynomials, inverse interpolation etc.)
  - Fourier Approximation
  - Case Studies : Curve Fitting
- **Numerical Differentiation & Integration**
  - Newton – Cotes Integration Formulas (trapezoidal rule, Simpson's rule, unequal segment etc.)
  - Integration of Equations

- Numerical Differentiation
- Case Studies: Numerical Differentiation & Integration

Module Code	**CS 2952**	**Module Title**	**Communication Skills**			
Credits	**1.5**	**Hours/Week**	**Lectures**	**0.5**	**Pre – requisites**	**None**
GPA/NGPA	**GPA**		**Lab/Assignments**	**3**		

Learning Outcomes

After completing this module, students should be able to

- express his/her ideas clearly in written format
- prepare technical documents depending on the target audience (technical or managerial)
- write summaries/concise articles from a given set of resources

Outline Syllabus

- **Writing newspaper articles**
- **Writing reports**
  - technical
  - non-technical (e.g. user documents)
- **Writing essays**
- **Critically evaluating a document**
  - explaining a point
  - agreeing to a point
  - disagreeing to a point
  - defending a decision
- **Writing summaries**

Module Code	CS 2032	Module Title	Principles of Computer Communication			
Credits	3	Hours/Week	Lectures	2	Pre – requisites	None
GPA/NGPA	GPA		Lab/Assignments	3		

Learning Outcomes

After completing this module, students should be able to

- explain the theoretical concepts behind data and voice communication
- analyse the application of communication principles in data and voice communication systems
- explain the use of communication principles in network applications

Outline Syllabus

- **Principles of Communications**
  - Frequency, Spectrum and Bandwidth
  - Analog and Digital data
  - Data encoding
  - Modulation
  - Transmission media
  - Transmission lines
  - Radio and microwave transmission
  - Satellite Transmission
  - Optical Fibre
  - Noise and transmission impairments
- **Transmission and Access networks**
  - Multiplexing and transmission hierarchies
  - DSL
  - Ethernet – media and topologies, standards, broadcast and switched, metro Ethernet
  - Wireless access networks – cellular, WLANs
- **Networking Principles**
  - Components of a Network
  - Circuit and packet switching
  - Convergence
- **Switching and routing**
  - Circuit switching
  - Packet switching and routing
  - The Internet Protocol (IP) – addressing and routing
- **Network applications**
  - Principles of network applications: client-server computing, e-mail, World wide web, web-2 applications, Telephony, SMS, MMS etc.
  - Video, Radio and TV broadcasting

Module Code	CS 2042	Module Title	Operating Systems			
Credits	2.5	Hours/Week	Lectures	2	Pre – requisites	CS1032
GPA/NGPA	GPA		Lab/Assignments	3/2		

Learning Outcomes

After completing this module, students should be able to

- illustrate concepts underlying design and implementation of contemporary operating systems, and chose and defend decisions on operating system implementation techniques
- develop system programs that simulate operating system behaviour
- apply knowledge about operating system behaviour to develop application programs
- evaluate decisions regarding selection of OSs for specific tasks

Outline Syllabus

- **Operating system as a virtual machine and a resource manager**
  - Discussion of various aspects of computer operating system implementations
  - Structure of a system call
- **Processes & Threads**
  - Process and Thread models
  - Inter-process communication and synchronization
  - Comparing Processes and Threads
- **Process and Thread scheduling**
  - Scheduling algorithms
  - Multi-processor scheduling
- **Deadlocks**
- **Memory management**
  - Memory model
  - Memory allocation and management techniques (e.g. :free list based management)
  - Segmentation
  - Paging
  - Page table operation
  - Page table design considerations
  - Virtual memory
  - Demand paging
  - Page replacement algorithms
  - Page fault handling
  - TLB implementation
- **IO management and Disk scheduling**
- **File systems**
  - Concept of files and directories
  - Directory structure
  - File system mounting
  - File system implementation

- Unix and Windows file systems

- **Protection and security**

Module Code	**CS 2062**	**Module Title**	**Object Oriented Software Development**			
Credits	**3**	**Hours/Week**	**Lectures**	**2**	**Pre – requisites**	**CS2012**
GPA/NGPA	**GPA**		**Lab/Assignments**	**3**		

Learning Outcomes

After completing this module, students should be able to

- develop clear, concise, and sufficiently formal requirements for extensions to an existing system, based on the true needs of users and other stakeholders
- apply design principles and patterns while designing and implementing systems based on reusable technology
- create UML class diagrams which model aspects of the domain and the software architecture
- create UML sequence diagrams and state machines that correctly model system behaviour

Outline Syllabus

- **Principles of software engineering**
  - Requirements specification and analysis, design, and testing
  - Reusable technologies as a basis for software engineering and use of frameworks and APIs
  - General software design principles
- **Object oriented analysis and design using UML**
  - Drawing UML class, package and component diagrams
  - User-centred design and use cases
  - Software behaviour representation with sequence diagrams, state machines, and activity diagrams
  - Design patterns
  - Evaluation of designs
- **Software architectures**
  - Domain-specific modelling
  - Architectural patterns

<table>
<tr><td>Module Code</td><td>CS 2202</td><td>Module Title</td><td colspan="4">Programming Challenge I</td></tr>
<tr><td>Credits</td><td>1</td><td rowspan="2">Hours/Week</td><td>Lectures</td><td></td><td rowspan="2">Pre – requisites</td><td rowspan="2">CS2012, CS2022</td></tr>
<tr><td>GPA/NGPA</td><td>NGPA</td><td>Lab/Assignments</td><td>3</td></tr>
<tr><td colspan="7"><u>Learning Outcomes</u>

After completing this module, students should be able to
• design an interactive software
• design a suitable algorithm to fulfill a given task
• plan a group oriented task
• demonstrate the skills of handling suitable software tools and technologies</td></tr>
<tr><td colspan="7"><u>Outline Syllabus</u>

• Designing software development projects
• Application domains
• Software development tools and technologies
• Algorithms
• Current trends in interactive programs</td></tr>
</table>

<table>
<tr><td>Module Code</td><td>CS 2962</td><td>Module Title</td><td colspan="4">Presentation Skills</td></tr>
<tr><td>Credits</td><td>1.5</td><td rowspan="2">Hours/Week</td><td>Lectures</td><td>0.5</td><td rowspan="2">Pre – requisites</td><td rowspan="2">None</td></tr>
<tr><td>GPA/NGPA</td><td>NGPA</td><td>Lab/Assignments</td><td>3</td></tr>
<tr><td colspan="7"><u>Learning Outcomes</u>

After completing this module, students should be able to
• present any given topic clearly through oral presentations
• demonstrate the ability to decide on the appropriate content for the presentation and the length of the presentation
• demonstrate the ability to engage the audience to the presentation
• review speeches and presentations in front of an audience</td></tr>
<tr><td colspan="7"><u>Outline Syllabus</u>

• Development of presentations
• Carrying out Q&A (Question and Answer) sessions based on the presentation
• “Ice-Breaker” sessions in front of the class
• Critical evaluation of presentations
• Debates
• Drama Competition
• Impromptu speeches, discussions
• Role playing</td></tr>
</table>

<table>
<tr><td>Module Code</td><td>CS 3022</td><td>Module Title</td><td colspan="4">Software Engineering</td></tr>
<tr><td>Credits</td><td rowspan="2">3</td><td rowspan="2">Hours/Week</td><td>Lectures</td><td>2</td><td rowspan="2">Pre – requisites</td><td rowspan="2">CS2062</td></tr>
<tr><td>GPA/NGPA</td><td>GPA</td><td>Lab/Assignments</td><td>3</td></tr>
<tr><td colspan="7"><u>Learning Outcomes</u>

After completing this module, students should be able to
• apply appropriate software engineering processes and techniques depending on the type and nature of the programming project.
• elicit and track formal requirements
• design and architect large scale software systems with appropriate levels of software and component reuse
• apply iterative software engineering principles to develop & deploy different types of software solutions with requisite quality within the budgeted cost and time</td></tr>
<tr><td colspan="7"><u>Outline Syllabus</u>

• Overview of software engineering & systems engineering
• Software processes
• Overview of project management
• Requirements engineering
• Requirements engineering processes
• System modeling
• Architectural design & documentation
• Architectural structure & styles
• User interface design
• Rapid software development & software reuse
• Overview of distributed systems & vertical enterprise applications, enterprise application integration & service-oriented architecture
• Overview of quality management and verification & validation
• Software testing</td></tr>
</table>

<table>
<tr><td>Module Code</td><td>CS 3032</td><td>Module Title</td><td colspan="4">Computer Networks</td></tr>
<tr><td>Credits</td><td>3</td><td rowspan="2">Hours/Week</td><td>Lectures</td><td>2</td><td rowspan="2">Pre – requisites</td><td rowspan="2">CS2032 or EN2052</td></tr>
<tr><td>GPA/NGPA</td><td>GPA</td><td>Lab/Assignments</td><td>3</td></tr>
<tr><td colspan="7"><u>Learning Outcomes</u>
After completing this module, students should be able to
• explain the different types of computer networks and the concepts behind them
• describe different networking protocols at different levels of protocol stack and relevant standards that define such protocols
• demonstrate the ability to set up networks given the requirements specification</td></tr>
<tr><td colspan="7"><u>Outline Syllabus</u>
• Introduction and OSI Model
• The Data Link Layer
o Data Link Layer concepts, protocols, functions and frame formats
o Error detection, correction & control
o ARQ
o Flow control
o Introduction to Token Ring, Token Bus, CSMA/CD etc.
• Wide-Area Networks (WANs)
o Leased Lines, Packet switching, X.25, Frame Relay, ATM, MPLS, Metro Ethernet
• IEEE LAN standards
o Introduction to IEEE 802.3/802.4/802.5 and related supplementary standards
• Network Layer
o Basis for specific implementations of network layers such as IP and other networking protocols
• Internet Protocol
o Details of Internet Protocol (datagram), Addressing, ICMP, Routing protocols, CIDR, Subnetting, ARP, DHCP, NAT,
o IPv6 and its headers, evolution and use of IPv6
• Transport Layer
o Transport layer functions, services, protocols, addressing & ports
o Client-server communications (connection establishment, data transfer & connection release), multiplexing, etc.
• Internet Transport Layer protocols
o TCP, UDP, header formats, ports and sockets
o TCP state machine
o Improvements to TCP
• Network Management
o OSI Management Model
o NW Management tools</td></tr>
</table>

- Simple Network Management Protocol

- **Network Design**
- **Other networking protocols**

Module Code	**CS 3042**	**Module Title**	**Database Systems**			
Credits	**3**	**Hours/Week**	**Lectures**	**2**	**Pre – requisites**	**None**
GPA/NGPA	**GPA**		**Lab/Assignments**	**3**		

Learning Outcomes

After completing this module, students should be able to

- explain the role of database systems in information management
- describe the basic fundamental concepts of data modelling and relational databases
- apply entity-relationship modelling and normalization for simple database requirements
- demonstrate the use of a query language to create, update and query a simple database
- construct simple applications that requires manipulating data in a DBMS

Outline Syllabus

- **Database Design Process & ER Model**
  - Database design and ER diagrams
  - Entities attributes and entity sets, Relationships and relationship sets, Keys and key constraints
  - Entity Vs relationship
  - Conceptual design for large enterprises
- **Relational Model**
  - Introduction to relational model
  - Integrity constraints over relations and enforcing integrity constraints
- **ER Model to Relational Model**
  - Entity sets to tables, Relationship sets to tables
  - Translating relationship sets with key constraints , s with participation constraints
  - Translating weak entities, class hierarchies and ER diagrams with aggregation
- **Normalization**
  - Attributes and anomalies
  - Loss-less join property, Dependency preserving property
  - 1st , 2nd and 3rd Normal Forms and Boyce-Codd Normal Form (BCNF)
- **Relational Algebra and Calculus**
  - Selection and projection
  - Set operations ,renaming, joins, division,
  - Tuple relational calculus, Domain relational calculus
- **Query Languages:**
  - Data Definition Language (DDL), Data Manipulation Language (DML), Data Query Language (DQL)
  - SQL Queries, Logical connectives , Security & authorization in SQL, Transaction processing using SQL, Embedding SQL in general purpose languages
- **Advanced SQL and Database Applications**
  - Embedded SQL, Cursors, Dynamic SQL

- JDBC, ODBC
- Stored procedures

- **Data Models**
  - External model, Conceptual model, Internal model and Physical model
- **Introduction to Indexes, Query Optimization, Physical Database Design**

Module Code	**CS 3072**	**Module Title**	**Micro-controllers and Applications**			
Credits	**3**	**Hours/Week**	**Lectures**	**2**	**Pre – requisites**	**CS2052**
GPA/NGPA	**GPA**		**Lab/Assignments**	**3**		

Learning Outcomes

After completing this module, students should be able to

- describe the concepts, architecture and limitations of single-chip computers and micro-controllers
- apply theory of micro-processor based design into real world applications
- select appropriate interfacing techniques for connecting digital systems to real world problems
- design, program and debug digital micro-controller based systems

Outline Syllabus

- **Introduction to micro-controllers, difference between micro-controller and microprocessors**
- **Micro-controller architecture**
- **I/O addressing, hand-shaking and timing diagrams**
- **Digital interfacing (Timers, Capture and Compare Units, Master & Salve ports)**
- **Analog interfacing – A/D and D/A converters, PWM converters**
- **Inter device communications – Computer Busses**
- **Micro-controller supervisory circuits**
- **Micro-controller based design techniques**
- **Micro-controller programming techniques**

<table>
<tr><td>Module Code</td><td>CS 2212</td><td>Module Title</td><td colspan="4">Programming Challenge II</td></tr>
<tr><td>Credits</td><td>1</td><td rowspan="2">Hours/Week</td><td>Lectures</td><td></td><td rowspan="2">Pre – requisites</td><td rowspan="2">CS2012, CS2022</td></tr>
<tr><td>GPA/NGPA</td><td>NGPA</td><td>Lab/Assignments</td><td>3</td></tr>
<tr><td colspan="7"><u>Learning Outcomes</u>

After completing this module, students should be able to
• develop interactive programs that have a high level of usability
• apply best coding practices
• contribute effectively for a group oriented task</td></tr>
<tr><td colspan="7"><u>Outline Syllabus</u>

• Software project management
• Software usability
• Game technologies
• GUI design
• Best coding practices</td></tr>
</table>

Module Code	CS 3952	Module Title	Technical Writing			
Credits	1	Hours/Week	Lectures	0.5	Pre – requisites	None
GPA/NGPA	NGPA		Lab/Assignments	1.5		

Learning Outcomes

After completing this module, students should be able to

- understand the important aspects of writing required in the world of technically-oriented professions
- develop the practical knowledge & capability to successfully communicate technical information in a manner adapted to the needs, levels of understanding and background of the readers/audience

Outline Syllabus

- **Document Design**
- **Technical Writing Process**
  - Audience analysis
  - Topic ideas, brainstorming, narrowing & outlining
  - Note taking & rough-drafting
  - Power-revision techniques
  - Libraries, documentation & cross-referencing
  - Strategies for peer-reviewing and team-writing
- **Technical Writing Guidelines**
  - Basic patterns and elements of a sentence
  - Common grammar, usage & punctuation problems
  - Common spelling problems
- **Technical Writing Scenarios & Applications**
  - Types of technical reports & different types of content (interface, reference, descriptive, conceptual and procedural information etc.)
  - Proposals
  - Progress Reports
  - Instructions
  - User Guides
  - Organizational policies & procedures
  - Feasibility, evaluation & recommendation reports
  - Technically-oriented articles (descriptive, definitive, causal, process, classification & comparison oriented communication)
  - Business plans
  - Writing abstracts, introductions & conclusions

<table>
<tr><td>Module Code</td><td>CS 3202</td><td>Module Title</td><td colspan="4">Software Engineering Project</td></tr>
<tr><td>Credits</td><td>2</td><td rowspan="2">Hours/Week</td><td>Lectures</td><td></td><td rowspan="2">Pre – requisites</td><td rowspan="2">CS2012, CS2020</td></tr>
<tr><td>GPA/NGPA</td><td>GPA</td><td>Lab/Assignments</td><td>6</td></tr>
<tr><td colspan="7"><u>Learning Outcomes</u>

After completing this module, students should be able to
• develop a software system
• apply software engineering processes and principles in developing software
• generate relevant software engineering documents
• abstract out important concepts in a project and write a technical paper
• develop software individually</td></tr>
<tr><td colspan="7"><u>Outline Syllabus</u>

Each student will have to work on an individual semester long programming project while rigorously applying software engineering principles using the latest source code management tools and leveraging specification templates used in the industry. These programming projects would typically entail projects where they have to add components or modify components in a large existing software project.

Students are required to perform the following during the course
• Understand an iterative software engineering process and the related documentation requirements
• Study the potential project idea and present a feasibility report
• Design a project plan with at least two iterations and appropriate milestones.
• Develop a project vision document
• Understand the overall software requirements and write the software requirements specification
• Analyze the requirements and create the software architecture and design
• Complete the development and demonstrate the project outcomes
• Extract the important concepts of the project and write a technical paper</td></tr>
</table>

Module Code	CS 3962	Module Title	Research & Report Writing			
Credits	1	Hours/Week	Lectures	0.5	Pre – requisites	None
GPA/NGPA	GPA		Lab/Assignments	1.5		

Learning Outcomes

After completing this module, students should be able to

- build and write an argument that would motivate the readers to accept their claims
- explain how to anticipate reader reservations and formulate appropriate responses accordingly
- write suitable introductions and conclusions in formal technical research reports
- demonstrate the practical ability to effectively plan, reason and write good reports

Outline Syllabus

- **Researcher as a writer**
  - The uses of research – public & private
  - Connecting with the reader
- **Planning a project**
- **Assembling a research argument**
  - Overview of making good arguments
  - Making & evaluating claims
  - Assembling reasons & evidence
  - Deciding acknowledgements and framing responses
  - Warrants
- **Components of a formal report**
- **Mechanics of writing styles**
- **Planning, drafting, writing and revising a report**
  - Common flaws in planning a report
  - Drafting a report
  - Quoting, paraphrasing, summarising & preventing accidental plagiarism
  - Thinking like a reader
  - Revising arguments and report organization
  - Communicating evidence visually
  - Introductions & conclusions
  - Principles of clear writing styles

<table>
<tr><td>Module Code</td><td>CS 3052</td><td>Module Title</td><td colspan="4">Computer Security</td></tr>
<tr><td>Credits</td><td>2</td><td rowspan="2">Hours/Week</td><td>Lectures</td><td>2</td><td rowspan="2">Pre – requisites</td><td rowspan="2">CS3022, CS 3032</td></tr>
<tr><td>GPA/NGPA</td><td>GPA</td><td>Lab/Assignments</td><td></td></tr>
<tr><td colspan="7"><u>Learning Outcomes</u>

After completing this module, students should be able to
• demonstrate the knowledge of fundamentals of computer and network security
• describe underlying theories and concepts of security
• apply access controls, solve Internet related security issues, and to take preventive and reactive measures to deal with computer security issues</td></tr>
<tr><td colspan="7"><u>Outline Syllabus</u>

• Introduction to security
• Features of security systems
• Threats and attacks on security
• Introduction to cryptography
• Digital signatures
• Program level security
• Computer system security
• Introduction to network security</td></tr>
</table>

<table>
<tr><td>Module Code</td><td>CS 3062</td><td>Module Title</td><td colspan="4">Theory of Computing</td></tr>
<tr><td>Credits</td><td>2</td><td rowspan="2">Hours/Week</td><td>Lectures</td><td>2</td><td rowspan="2">Pre – requisites</td><td rowspan="2">CS2022</td></tr>
<tr><td>GPA/NGPA</td><td>GPA</td><td>Lab/Assignments</td><td></td></tr>
<tr><td colspan="7"><u>Learning Outcomes</u>

After completing this module, students should be able to
• explain and distinguish the basic models of computation and the Chomsky-hierarchy
• describe the relationship between basic models and corresponding languages
• explain the limitations of basic models and what happens when enhancements (e.g., non-determinism, multiple tapes) are introduced
• describe a Turing Machine and the Church-Turing thesis</td></tr>
<tr><td colspan="7"><u>Outline Syllabus</u>

• Overview of Basic models of computation
• Finite automata and regular languages
• Pushdown automata and context-free languages
• Turing machines and their languages
• Church-Turing thesis
• Complexity, Computability and non-computability</td></tr>
</table>

Module Code	CS 3212	Module Title	Software Architecture and Design			
Credits	3	Hours/Week	Lectures	2	Pre – requisites	None
GPA/NGPA	GPA		Lab/Assignments	3		

Learning Outcomes

After completing this module, students should be able to

- evaluate a software architecture with respect to strategic vision and goals of an enterprise
- use a range of software design patterns, frameworks and architectures for design of software
- select and apply appropriate metrics to evaluate qualities of a software design including performance, reliability, security, safety, re-usability, complexity, etc.

Outline Syllabus

- **Principles of software architecture**
  - Mapping enterprise strategic vision to architectural modal
  - Architectural modelling through views
  - Evaluation of architectures, design patterns and application frameworks
- **Principles of software design**
  - Modularization of functionality and elicitation of system properties
  - Design strategies and evaluation of designs
  - Cross-cutting concerns in a software design
- **Practice of software architecture and design**
  - Design of distributed systems, component based design and software as a service
  - Documenting software architectures
  - Reuse of architectures
  - Case studies in software architecture and design

<table>
<tr><td>Module Code</td><td>CS 3312</td><td>Module Title</td><td colspan="4">Embedded System Design</td></tr>
<tr><td>Credits</td><td>3</td><td rowspan="2">Hours/Week</td><td>Lectures</td><td>2</td><td rowspan="2">Pre – requisites</td><td rowspan="2">CS2042, EN1022, CS 2052</td></tr>
<tr><td>GPA/NGPA</td><td>GPA</td><td>Lab/Assignments</td><td>3</td></tr>
<tr><td colspan="7"><u>Learning Outcomes</u>

After completing this module, students should be able to
• explain the design concepts of embedded systems and “System On Chip” designs
• model real world systems into embedded systems
• design, model and simulate embedded systems
• explain the design and implementation constraints of embedded systems
• explain the economics of embedded systems design and implementation</td></tr>
<tr><td colspan="7"><u>Outline Syllabus</u>

• Introduction to embedded applications and “System on Chip” implementations
• High-level modelling of real world systems (Black box models and glass-box models)
• System level models (Data-flow, Behavioural and architectural models)
• Hardware – software co-design
• Providing real-time response
• Error handling, fail-safe and supervisory techniques
• Power management
• Simulation, prototyping and testing methods for embedded applications</td></tr>
</table>

<table>
<tr><td>Module Code</td><td>CS 3412</td><td>Module Title</td><td colspan="4">Networking Technologies</td></tr>
<tr><td>Credits</td><td>3</td><td rowspan="2">Hours/Week</td><td>Lectures</td><td>2</td><td rowspan="2">Pre – requisites</td><td rowspan="2">CS3032</td></tr>
<tr><td>GPA/NGPA</td><td>GPA</td><td>Lab/Assignments</td><td>3</td></tr>
<tr><td colspan="7"><u>Learning Outcomes</u>

After completing this module, the students should be able to
• demonstrate the knowledge of the operation of current networking technologies

• show the application of different networking technologies in implementing networks</td></tr>
<tr><td colspan="7"><u>Outline Syllabus</u>

• Theory and implementation of the following technologies, and any other new technology introduced from time to time:
o Wireless networking technologies – WiFi, WiMax, 3G/4G
o DSL technologies
o Metro Ethernet
o Frame Relay, ATM
o Fibre Channel
o EFM, FTTx
o MPLS
o VLANs, VPNs
o SANs</td></tr>
</table>

<table>
<tr><td>Module Code</td><td>CS 3512</td><td>Module Title</td><td colspan="4">Programming Languages</td></tr>
<tr><td>Credits</td><td>3</td><td rowspan="2">Hours/Week</td><td>Lectures</td><td>2</td><td rowspan="2">Pre – requisites</td><td rowspan="2">CS2022</td></tr>
<tr><td>GPA/NGPA</td><td>GPA</td><td>Lab/Assignments</td><td>3</td></tr>
<tr><td colspan="7"><u>Learning Outcomes</u>

After completing this module, the students should be able to
• understand the principles behind the design of programming languages and compilers
• compare different programming paradigms</td></tr>
<tr><td colspan="7"><u>Outline Syllabus</u>

• Compilation and Interpretation Concepts
o Syntax specification, scanning & parsing
o Names, scopes and bindings
o Semantic analysis
o Target Machine Architecture
• Programming Language Design Issues
o Control flow
o Data types
o Subroutines and control abstraction
o Data abstraction and object orientation
• Alternative Programming Models
o Functional languages - Scheme
o Logic languages - Prolog
o Concurrent Programming
o Scripting Languages
• Programming Language Implementation</td></tr>
</table>

- Compiler structure, intermediate code, code generation, assembly & linking
- Runtime program management
- Code improvement

<table>
<tr><td>Module Code</td><td>CS 3612</td><td>Module Title</td><td colspan="4">Intelligent Systems</td></tr>
<tr><td>Credits</td><td>3</td><td rowspan="2">Hours/Week</td><td>Lectures</td><td>2</td><td rowspan="2">Pre – requisites</td><td rowspan="2">CS2022</td></tr>
<tr><td>GPA/NGPA</td><td>GPA</td><td>Lab/Assignments</td><td>3</td></tr>
<tr><td colspan="7"><u>Learning Outcomes</u>

After completing this module, students should be able to
• recognise the unique problems which AI attempts to solve
• explain underlying issues and commonly used AI techniques and strategies adopted to create intelligent programs
• analyse and select the most efficient and appropriate mechanism applicable to a given problem that can be solved using AI</td></tr>
<tr><td colspan="7"><u>Outline Syllabus</u>

• Introduction and Fundamental Issues
• Intelligent agents
• Intelligent search techniques
• Games as search problems
• Learning and Knowledge Representation
• Language and programming techniques for AI
• Artificial planning and learning techniques</td></tr>
</table>

<table>
<tr><td>Module Code</td><td>CS 3712</td><td>Module Title</td><td colspan="4">Image Processing</td></tr>
<tr><td>Credits</td><td>3</td><td rowspan="2">Hours/Week</td><td>Lectures</td><td>2</td><td rowspan="2">Pre – requisites</td><td rowspan="2">CS2022</td></tr>
<tr><td>GPA/NGPA</td><td>GPA</td><td>Lab/Assignments</td><td>3</td></tr>
<tr><td colspan="7"><u>Learning Outcomes</u>

After completing this module, students should be able to
• describe the human image perception mode
• apply qualitative and quantitative image fidelity criterions
• describe types of noise in images and noise removal techniques
• describe and apply image quality enhancement techniques
• describe principles of image compression models and apply image compression techniques
• apply principles of digital video</td></tr>
<tr><td colspan="7"><u>Outline Syllabus</u>

• Introduction to human vision system
• Image fidelity criterions
• Digital image acquisition methods
• Point and neighbourhood operations
• Types of noise and noise removal techniques
• Image transforms and enhancement
• Image compression (lossy and lossless compression)
• GIF and JPEG standards
• Principles of digital video
• MPEG video compression standard</td></tr>
</table>

Module Code	CS 3992	Module Title	Industrial Training			
Credits	6.0	Hours/Week	Lectures	-	Pre – requisites	None
GPA/NGPA	NGPA		Lab/Assignments	-		

Learning Outcomes

After completing this module, the students should be able to

- explain the role and contribution of industry for the national development
- recognize the professional realities of the industrial environment that complement, enhance, confirm, and reshape knowledge
- appreciate the formal and informal relationships in an industrial organization and demonstrate essential skills on human relations and team work
- demonstrate the ability to appreciate the complex nature of industrial problems and relate theoretical concepts learnt to practices in the industry in obtaining optimum solutions.
- appreciate the application and synergy of engineering, management, financial and economic aspects in the real world.
- demonstrate the ability to cultivate powers of observation and make use of the facilities provided to fulfil the needs of the industry.
- demonstrate the sense of responsibility towards industry and society in general

Outline Syllabus

- **Orientation/Familiarization with the Industrial Training Organization**
  - Overall function/role/scope of the organization in the respective sector
  - Scale/magnitude of the organization in the respective sector in the country/region
  - Structure of the organization with respective to the discharge of different functions
  - Nature and scope of different engineering disciplines in respective divisions/departments.
- **Planning & designing of engineering products, process and projects and effecting.**
  - Research and concept development
  - Process of matching the client needs, equipment/component/device/ selection, tender procedures and analyse the requirements of the system or software
  - Prepare the design and the specification and evaluation of technical design details with respect to planning process based on need of the clients.
  - Managing and planning a Project
  - Costing and financing
  - Forecasting of human resource requirement and development
  - Setting up Disaster recovery techniques
  - Assuring Data protection and analyse the security and other risks.
  - Coding to implement the system
  - Testing and assuring the quality of the system
  - Prepare documentation on System or Application Software and user manuals.
  - Familiarize with systems
  - Commissioning procedures
  - Installation and testing
  - Use of tools

- Installing System / Network installation
- Managing System / Network and Troubleshooting.

- **Skills of functioning as an effective Engineer in the Industry**
  - Acquisition of professional ethics in an industry set up (transition from academic life to industry set up)
  - Working with non – engineers ranging from labourers to Chief Executives possessing vastly different education
  - Social and cultural experiences and backgrounds and becoming a team player and a leader
  - Earning the respect of all by demonstrating the engineering knowledge appropriately
  - Management, organization and labour rules and regulations.

<table>
<tr><td>Module Code</td><td>CS 4202</td><td>Module Title</td><td colspan="4">Research and Development Project</td></tr>
<tr><td>Credits</td><td>10</td><td rowspan="2">Hours/Week</td><td>Lectures</td><td>-</td><td rowspan="2">Pre – requisites</td><td rowspan="2">CS3202, CS3212</td></tr>
<tr><td>GPA/NGPA</td><td>GPA</td><td>Lab/Assignments</td><td>6</td></tr>
<tr><td colspan="7"><u>Learning Outcomes</u>

After completing this module, the students should be able to demonstrate
• the knowledge gained through research

• the skills in planning, documenting, project management and teamwork

• the developed project</td></tr>
<tr><td colspan="7"><u>Outline Syllabus</u>

• Project requirement gathering
• Preparation of project requirement specifications
• Project planning
• Software, hardware or system design (depending on the project)
• Acquiring domain knowledge through literature survey or by working with the industry
• Researching on current and past development and research work done by others
• Development of necessary skills for project implementation
• Prototyping the system
• Development of the system using the skills acquired
• Validation and testing of the development
• Periodic reporting and presentations
• Project report writing
• Research paper writing</td></tr>
</table>

Module Code	CS 4222	Module Title	Software Process and Management			
Credits	3	Hours/Week	Lectures	2	Pre – requisites	None
GPA/NGPA	GPA		Lab/Assignments	3		

Learning Outcomes

After completing this module, students should be able to

- evaluate frameworks for managing the quality of software work processes and software products with respect to target processes and products
- analyse methodologies for managing software projects and design customized plans for project management
- develop a comprehensive project plan based on standards for software process management

Outline Syllabus

- **Principles of software processes**
  - Relationship between software requirements evolution and process frameworks
  - Software process concepts and models
  - Process implementation
- **Software process quality**
  - Quality concepts and cultural aspects
  - Standards for quality of software processes an d modelling of cost of quality
  - Metrics for evaluation of quality
- **Project management**
  - Management models and impact on software processes
  - Project planning for resource allocation and risk management
  - Project control through measurement and analysis of results

Module Code	CS 4232	Module Title	Formal Methods in Software Engineering			
Credits	3	Hours/Week	Lectures	2	Pre – requisites	None
GPA/NGPA	GPA		Lab/Assignments	3		

Learning Outcomes

After completing this module, students should be able to

- create mathematically precise specifications and design for software using formal languages
- analyse the properties of formal specifications and designs for evaluation of quality and validation for compliance with requirements
- use tools to transform specifications and designs

Outline Syllabus

- **Concepts of formal methods**
  - Formal methods as a paradigm for higher quality software design and construction
  - Mathematical foundations, logic calculi and automata theory
  - Proofs of correctness for specifications and designs using formal methods
  - Validation of a formal model against implementable systems
- **Practice of formal models**
  - Types of models, formal languages, type systems and program semantics
  - Automated proof systems using theorem proving and model checking
  - Transformation of programs for formal analysis and verification
  - Analysis of software using formal methods
- **Integration of formal methods to software engineering processes**
  - Formal methods use in requirements analysis, refinement and testing
  - Reusable models and theories

<table>
<tr><td>Module Code</td><td>CS4322</td><td>Module Title</td><td colspan="4">Digital Systems Design</td></tr>
<tr><td>Credits</td><td>3</td><td rowspan="2">Hours/Week</td><td>Lectures</td><td>2</td><td rowspan="2">Pre – requisites</td><td rowspan="2">EN2022</td></tr>
<tr><td>GPA/NGPA</td><td>GPA</td><td>Lab/Assignments</td><td>3</td></tr>
<tr><td colspan="7"><u>Learning Outcomes</u>

After completing this module, students should be able to
• design complex constrained digital systems using modular MSI/LSI components
• understand and describe modern digital design techniques such as architectural state machine design and the use of re-configurable design blocks
• use hardware description language VHL modelling and synthesis of digital systems</td></tr>
<tr><td colspan="7"><u>Outline Syllabus</u>

• Structured design concept. Design strategies. Design decomposition. And design with architectural components.
• Introduction to VHDL
• Basic VHDL modeling techniques.
• Data flow modelling
• Behavioural modelling
• Structural modelling
• Design synthesis and verification
• Digital design project.</td></tr>
</table>

<table>
<tr><td>Module Code</td><td>CS 4342</td><td>Module Title</td><td colspan="4">Advanced Computer Architecture</td></tr>
<tr><td>Credits</td><td>3</td><td rowspan="2">Hours/Week</td><td>Lectures</td><td>2</td><td rowspan="2">Pre – requisites</td><td rowspan="2">CS2052 or EN2022</td></tr>
<tr><td>GPA/NGPA</td><td>GPA</td><td>Lab/Assignments</td><td>3</td></tr>
<tr><td colspan="7"><u>Learning Outcomes</u>

After completing this module, students should be able to
• provide comprehensive detailed explanations of the design concepts and considerations behind today’s microprocessors
• analyze the current and potential future development in computer systems technology
• critically evaluate the design concepts of modern multiprocessor systems
• relate operations of compilers and applications to processor architectures</td></tr>
<tr><td colspan="7"><u>Outline Syllabus</u>

• Fundamental Concepts
• Simple Pipelined RISC Processors
• Extending Simple Pipelined RISC Processors
• ILP and Superscalar processors
• Extracting ILP with Compiler Support
• Memory Hierarchy
• Multiprocessors and Thread Level Parallelism (TLP)
• Input/Output and Storage Systems</td></tr>
</table>

<table>
<tr><td>Module Code</td><td>CS 4422</td><td>Module Title</td><td colspan="4">Network Design</td></tr>
<tr><td>Credits</td><td>3</td><td rowspan="2">Hours/Week</td><td>Lectures</td><td>2</td><td rowspan="2">Pre – requisites</td><td rowspan="2">CS3032, CS3412</td></tr>
<tr><td>GPA/NGPA</td><td>GPA</td><td>Lab/Assignments</td><td>3</td></tr>
<tr><td colspan="7"><u>Learning Outcomes</u>

After completing this module, students should be able to
• design a large scale network using appropriate networking technologies and design parameters, to fulfil the requirements of an organization</td></tr>
<tr><td colspan="7"><u>Outline Syllabus</u>

• Analysis of organizational requirements for networking
• Network Design parameters
o Topology
o Security
o Performance
o Function
• Financial analysis, maintainability, usability, expandability
• Network design project – right throughout the semester</td></tr>
</table>

<table>
<tr><td>Module Code</td><td>CS4432</td><td>Module Title</td><td colspan="4">Network and System Management</td></tr>
<tr><td>Credits</td><td>3</td><td rowspan="2">Hours/Week</td><td>Lectures</td><td>2</td><td rowspan="2">Pre – requisites</td><td rowspan="2">None</td></tr>
<tr><td>GPA/NGPA</td><td>GPA</td><td>Lab/Assignments</td><td>3</td></tr>
<tr><td colspan="7"><u>Learning Outcomes</u>

After completing this module, the students should be able to
• configure networks and systems according to recommended guidelines
• manage users in a system
• manage systems
• manage networks
• manage and maintain services</td></tr>
<tr><td colspan="7"><u>Outline Syllabus</u>

• System configuration and maintenance – recommendations and best practices
• Network management – fault management, configuration, accounting, performance, security
• System management – fault management, configuration, accounting, performance, security
• Storage management – storage, backups, redundancy,
• Services management – licensing, distributed services, multi-tier applications
• Disaster recovery and business continuity management
• Policy frameworks and enforcement – use of networks and systems in a business environment</td></tr>
</table>

Module Code	CS4522	Module Title	Advanced Algorithms			
Credits	3	Hours/Week	Lectures	2	Pre – requisites	CS2022
GPA/NGPA	GPA		Lab/Assignments	3		

Learning Outcomes

After completing this module, students should be able to

- modify basic data structures to overcome drawbacks
- apply advanced techniques for analysis of algorithms
- apply advanced techniques for design of algorithms
- design and analyze algorithms in selected/specialized domains

Outline Syllabus

- **Advanced data structures**
  - Red-black trees
  - Binary decision diagrams (BDDs)
  - B-trees
  - Splay trees
- **Advanced design and analysis**
  - Randomized algorithms
  - Amortized analysis
  - Optimization techniques
- **Graph algorithms**
  - Flow networks, max-flow-min-cut theorem
  - All-pairs-shortest-path problem, transitive closure
- **Matrix and numerical algorithms**
- **Other /current topics**

<table>
<tr><td>Module Code</td><td>CS4532</td><td>Module Title</td><td colspan="4">Concurrent Programming</td></tr>
<tr><td>Credits</td><td>3</td><td rowspan="2">Hours/Week</td><td>Lectures</td><td>2</td><td rowspan="2">Pre – requisites</td><td rowspan="2">CS2012, CS2060, CS2042</td></tr>
<tr><td>GPA/NGPA</td><td>GPA</td><td>Lab/Assignments</td><td>3</td></tr>
<tr><td colspan="7"><u>Learning Outcomes</u>

After completing this module, students should be able to
• explain the concurrency aspects of a specified system
• evaluate methods of implementing concurrency for a given situation
• evaluate the impact of concurrency implementation on system components
• implement systems with concurrency</td></tr>
<tr><td colspan="7"><u>Outline Syllabus</u>

• Theory of concurrency
o Shared objects and Mutual exclusion
o Registers
o Linearization
o Consensus
o Lock free synchronization
o Wait free synchronization
o Safety and liveness properties
• Synchronisation methods
o Semaphores
o Monitors
o Barriers
o Spin locks
o Compare-and Set
• Parallel programming with multiprocessors
• Cache optimization
• Concurrent data Structures
o Lock-free and wait free stacks and queues</td></tr>
</table>

<table>
<tr><td>Module Code</td><td>CS4542</td><td>Module Title</td><td colspan="4">Compiler Design</td></tr>
<tr><td>Credits</td><td>3</td><td rowspan="2">Hours/Week</td><td>Lectures</td><td>2</td><td rowspan="2">Pre – requisites</td><td rowspan="2">None</td></tr>
<tr><td>GPA/NGPA</td><td>GPA</td><td>Lab/Assignments</td><td>3</td></tr>
<tr><td colspan="7"><u>Learning Outcomes</u>

After completing this module, students should be able to
• explain the stages of a compiler and how they operate
• apply common techniques and tools used in designing compilers
• analyze compiler implementation techniques and simple optimization techniques</td></tr>
<tr><td colspan="7"><u>Outline Syllabus</u>

• Review of lexical, syntax and semantic analysis
• Implementation techniques and run-time issues
• Intermediate representation
• Supporting language features
• Code generation and code optimization techniques
• Data flow analysis
• Control flow analysis
• Basic blocks
• Liveness analysis
• Inter-procedural analysis
• Machine dependant optimisation</td></tr>
</table>

<table>
<tr><td>Module Code</td><td>CS4632</td><td>Module Title</td><td colspan="4">Database Internals</td></tr>
<tr><td>Credits</td><td>3</td><td rowspan="2">Hours/Week</td><td>Lectures</td><td>2</td><td rowspan="2">Pre – requisites</td><td rowspan="2">CS3042</td></tr>
<tr><td>GPA/NGPA</td><td>GPA</td><td>Lab/Assignments</td><td>3</td></tr>
<tr><td colspan="7"><u>Learning Outcomes</u>

After completing this module, the students should be able to
• explain the basic ACID properties of a database
• describe the basic fundamental concepts in transaction processing
• apply the knowledge of database internals to improve the performance of database systems
• demonstrate the use of an appropriate database solution for an application that requires complex data storage and manipulation</td></tr>
<tr><td colspan="7"><u>Outline Syllabus</u>

• File Organization and Indexes
• Query Optimization
• Physical Database Design tuning decisions
• Transaction Management in Databases
• Database Administration
• Parallel and Distributed Databases
• Active Databases
• Temporal Databases
• Spatial, Text and Multimedia Databases
• Object Databases
• Deductive Databases
• Current applications and research trends in advanced database topics</td></tr>
</table>

<table>
<tr><td>Module Code</td><td>CS4642</td><td>Module Title</td><td colspan="4">Data Mining and Information Retrieval</td></tr>
<tr><td>Credits</td><td>3</td><td rowspan="2">Hours/Week</td><td>Lectures</td><td>2</td><td rowspan="2">Pre – requisites</td><td rowspan="2">CS3612</td></tr>
<tr><td>GPA/NGPA</td><td>GPA</td><td>Lab/Assignments</td><td>3</td></tr>
<tr><td colspan="7"><u>Learning Outcomes</u>

After completing this module, the students should be able to
• describe the basic fundamental concepts involved in the process of discovering useful, possibly unexpected, patterns in large data sets
• explain the various stages involved in the data mining and information retrieval process
• explain the computational challenges and appropriate solutions available in implementing modern data mining and information retrieval systems
• evaluate, choose and implement an appropriate solution for a given data mining or/and information retrieval application from a wide array of application domains</td></tr>
<tr><td colspan="7"><u>Outline Syllabus</u>

• Introduction and motivation
• Data mining primitives including data representation
• Data pre-processing
• Data warehousing
• Data visualization
• Concept descriptions
• Association rule mining
• Classification
• Clustering
• Page ranking and searching
• Ontology engineering</td></tr>
</table>

<table>
<tr><td>Module Code</td><td>CS4722</td><td>Module Title</td><td colspan="4">Computer Vision</td></tr>
<tr><td>Credits</td><td>2</td><td rowspan="2">Hours/Week</td><td>Lectures</td><td>2</td><td rowspan="2">Pre – requisites</td><td rowspan="2">None</td></tr>
<tr><td>GPA/NGPA</td><td>3</td><td>Lab/Assignments</td><td>3</td></tr>
<tr><td colspan="7"><u>Learning Outcomes</u>

After completing this module, students should be able to
• explain the principles of machine vision and how it can be used in practical applications
• judge the importance of machine perception of images
• apply mathematical techniques and related algorithms for image understanding</td></tr>
<tr><td colspan="7"><u>Outline Syllabus</u>

• Image formation
• Edge detection
• Boundary detection and image segmentation
• Image features
• Mathematical morphology
• Shape and boundary descriptors
• Elementary pattern recognition
• Texture recognition
• Marr's paradigm
• Active vision
• Calibration problems
• Correspondence and flow
• Stereo and motion</td></tr>
</table>

<table>
<tr><td>Module Code</td><td>CS4732</td><td>Module Title</td><td colspan="4">Computer Graphics</td></tr>
<tr><td>Credits</td><td>3</td><td rowspan="2">Hours/Week</td><td>Lectures</td><td>2</td><td rowspan="2">Pre – requisites</td><td rowspan="2">None</td></tr>
<tr><td>GPA/NGPA</td><td>GPA</td><td>Lab/Assignments</td><td>3</td></tr>
<tr><td colspan="7"><u>Learning Outcomes</u>

After completing this module, the students should be able to
• demonstrate the difference among different types of computer graphic representation
• select suitable graphic formats and transforms for different application domains
• create and edit graphic elements and content using different types of tools
• publish graphical media on different rendering platforms</td></tr>
<tr><td colspan="7"><u>Outline Syllabus</u>

• Introduction to computer graphics
• Types of graphic content; raster Vs vector, linear Vs non-linear graphics
• Rendering techniques
• 2D and 3D graphics
• Graphic and media publishing platforms
• Projections and transforms
• Illumination and shading
• Texture creation and mapping
• Principles of animation</td></tr>
</table>

Module Code	CS4742	Module Title	Bioinformatics			
Credits	3	Hours/Week	Lectures	2	Pre – requisites	None
GPA/NGPA	GPA		Lab/Assignments	3		

Learning Outcomes

After completing this module, students should be able to

- appreciate, understand and develop applications of computer technology to management of biological information
- apply algorithms to facilitate and expedite biological research in analysing and visualising vast amounts of genome data that are being gathered presently

Outline Syllabus

- **Molecular Biology**
  - An introduction to Bioinformatics
  - Molecular biology
- **Biological Sequence Data Analysis**
  - Basic alignment problems
  - Database searches
  - Heuristic alignment algorithms
- **Pattern Discovery and Recognition**
  - Gene finding and Motif discovery
  - Hidden Markov models
  - Sequence profiles
- **Systems biology**
  - Biological networks
  - Network Motif analysis
- **Applications of bioinformatics**

<table>
<tr><td>Module Code</td><td>CS4012</td><td>Module Title</td><td colspan="4">Professional Practice</td></tr>
<tr><td>Credits</td><td>2</td><td rowspan="2">Hours/Week</td><td>Lectures</td><td>2</td><td rowspan="2">Pre – requisites</td><td rowspan="2">None</td></tr>
<tr><td>GPA/NGPA</td><td>GPA</td><td>Lab/Assignments</td><td>0</td></tr>
<tr><td colspan="7"><u>Learning Outcomes</u>

After completing this module, students should be able to
• understand and analyze social and professional issues that arise in the practice of computing</td></tr>
<tr><td colspan="7"><u>Outline Syllabus</u>

• Engineering ethics and value systems
• Technology-driven changes to public policy
• Technology, law and the working environment
• Law and intellectual properties
• Community standards and personal responsibility</td></tr>
</table>

<table>
<tr><td>Module Code</td><td>CS4242</td><td>Module Title</td><td colspan="4">Human-Computer Interaction</td></tr>
<tr><td>Credits</td><td>3</td><td rowspan="2">Hours/Week</td><td>Lectures</td><td>2</td><td rowspan="2">Pre – requisites</td><td rowspan="2">CS2012, CS2020</td></tr>
<tr><td>GPA/NGPA</td><td>GPA</td><td>Lab/Assignments</td><td>3</td></tr>
<tr><td colspan="7"><u>Learning Outcomes</u>

After completing this module, the students should be able to
• explain why good HCI design is needed
• identify the relevant characteristics of humans and computers
• demonstrate a knowledge of HCI concepts
• design high quality user interfaces</td></tr>
<tr><td colspan="7"><u>Outline Syllabus</u>

• Relevant characteristics of humans and computers
• The interaction between humans and systems
• Interaction design
• HCI in the software process
• Universal design and localisation
• User support
• HCI Design Project</td></tr>
</table>

<table>
<tr><td>Module Code</td><td>CS4252</td><td>Module Title</td><td colspan="4">Advanced Operating Systems</td></tr>
<tr><td>Credits</td><td>3</td><td rowspan="2">Hours/Week</td><td>Lectures</td><td>2</td><td rowspan="2">Pre – requisites</td><td rowspan="2">CS2042</td></tr>
<tr><td>GPA/NGPA</td><td>GPA</td><td>Lab/Assignments</td><td>3</td></tr>
<tr><td colspan="7"><u>Learning Outcomes</u>

After completing this module, the students should be able to
• explain design strategies required to provide functionality beyond basic OS services

• critically evaluate a specified advanced operating system
• synthesise an evaluation criteria for an OS targeted for a non-conventional environment</td></tr>
<tr><td colspan="7"><u>Outline Syllabus</u>

• Design concepts and features of different operating systems
• Advanced functions of operating systems
• Synchronization and communication
• File systems design
• Protection and security
• Operating system structure and extension techniques
• Fault tolerance
• Distributed operating systems</td></tr>
</table>

<table>
<tr><td>Module Code</td><td>CS4262</td><td>Module Title</td><td colspan="4">Distributed Systems</td></tr>
<tr><td>Credits</td><td>3</td><td rowspan="2">Hours/Week</td><td>Lectures</td><td>2</td><td rowspan="2">Pre – requisites</td><td rowspan="2">None</td></tr>
<tr><td>GPA/NGPA</td><td>GPA</td><td>Lab/Assignments</td><td>3</td></tr>
<tr><td colspan="7"><u>Learning Outcomes</u>

After completing this module, students should be able to
• understand the nature & characteristics of distributed systems
• identify how and when to apply particular distributed algorithms and techniques
• understand the latest distributed computing technology & application trends</td></tr>
<tr><td colspan="7"><u>Outline Syllabus</u>

• Characterization of distributed systems
• Distributed system architectural and fundamental models
• Network communication
• Communication models
• Distributed computing applications
• Distributed computing middleware
• Time, synchronization and global states
• Coordination and agreement
• P2P systems and grid computing
• Transactions and concurrency control
• Consistency and replication</td></tr>
</table>

Module Code	CS4272	Module Title	Quality Engineering			
Credits	3	Hours/Week	Lectures	2	Pre – requisites	None
GPA/NGPA	GPA		Lab/Assignments	3		

Learning Outcomes

After completing this module, students should be able to

- understand quality management process activities such as quality assurance, quality planning and quality control
- understand how measurements help in assessing quality attributes and the differences between predictor and control metrics
- distinguish between software verification and validation techniques and between validation testing and defect testing
- understand the principles of system testing and component testing

Outline Syllabus

- **Principles of software verification and validation**
- **Software quality metrics**
- **Project quality management**
  - Defect management process, defect removal effectiveness
- **Software inspection and testing**
- **Fundamentals**
  - Concepts and context, risk analysis, goals, process, and psychology
- **Lifecycle testing**
  - How testing relates to development including models, verification and validation, and types of tests
- **Static testing**
  - Reviews, inspections, and static tools
- **Non-functional testing**
- **System and component testing**
- **Test design techniques**
  - Black-box test methods, white-box techniques, error guessing, and exploratory testing
- **Test management**
  - Team organization, key roles and responsibilities, test strategy and planning, test execution documentation, configuration management
- **Testing tools**
  - Tool selection, benefits, risks, and classifications
- **Continuous integration and automated testing**
- **Cost of software quality**
- **Maturity models for software test process assessment and improvement**

<table>
<tr><td>Module Code</td><td>CS4282</td><td>Module Title</td><td colspan="4">Mobile Computing</td></tr>
<tr><td>Credits</td><td>3</td><td rowspan="2">Hours/Week</td><td>Lectures</td><td>2</td><td rowspan="2">Pre – requisites</td><td rowspan="2">CS3022, CS3312</td></tr>
<tr><td>GPA/NGPA</td><td>GPA</td><td>Lab/Assignments</td><td>3</td></tr>
<tr><td colspan="7"><u>Learning Outcomes</u>

After completing this module, the students should be able to
• identify the technologies, devices and applications which enable mobile computing
• discover the issues which need to be considered when developing a mobile application
• design and implement sophisticated mobile applications</td></tr>
<tr><td colspan="7"><u>Outline Syllabus</u>

• Mobility in computing systems
• Mobility management
• Mobile and wireless networking technologies and standards
• Data and resource management
• Design issues for mobile computing systems
• Technologies used to enable mobility in information systems
• Design and implementation of mobile applications
• Further topics</td></tr>
</table>

Module Code	CS4332	Module Title	Computer Aided Digital Design			
Credits	3	Hours/Week	Lectures	2	Pre – requisites	CS3612
GPA/NGPA	GPA		Lab/Assignments	3		

Learning Outcomes

After completing this module, students should be

- able to identify why computer-aided design is necessary and its impact on the design process of modern digital systems
- familiar with some commonly used computer-aided design tools, their main features/capabilities and limitations
- aware of the basic concepts about hardware description languages (HDLs)
- able to model and synthesize digital systems using HDLs

Outline Syllabus

- **Architectural and modular modelling of digital systems**
  - PLA and PAL architectures
  - LUT based synthesis / modelling of digital systems
  - Synchronous and Asynchronous systems
  - ASM notation for State machine based modelling
- **Advanced programmable logic devices**
  - Introductions to structures of CPLD and FGPA devices
  - ASICs
- **Introduction to VHDL**
  - Capabilities, Terminology and Synthesis process engine (Design Entry, Synthesis, Implementation, Timing and Functional verification)
  - Introduction to Xilinx VHDL design environment
- **VHDL Elements**
  - Data objects, types & operators
  - Concurrent and sequential statements Styles of modelling
  - Behavioural, data flow and structural modelling of Digital Systems.
- **Logic-level Synthesis and Optimization**
  - Two-level logic optimization: exact and approximate methods
  - Binary Decision Diagrams
  - Multi-level logic optimization: methods and tools
  - Sequential logic: synthesis and optimization
- **Testing of digital systems and design for testability**

<table>
<tr><td>Module Code</td><td>CS4352</td><td>Module Title</td><td colspan="4">Robotics and Automation</td></tr>
<tr><td>Credits</td><td>3</td><td rowspan="2">Hours/Week</td><td>Lectures</td><td>2</td><td rowspan="2">Pre – requisites</td><td rowspan="2">CS3242</td></tr>
<tr><td>GPA/NGPA</td><td>GPA</td><td>Lab/Assignments</td><td>3</td></tr>
<tr><td colspan="7"><u>Learning Outcomes</u>

After completing this module, students should be able to
• explain the relationship between mechanical structures and their operational workspace characteristics
• apply spatial transformation to obtain forward kinematics equation of robot manipulators
• solve inverse kinematics of simple robot manipulators and generate joint trajectory for motion
• evaluate robot controllers for the purpose of interfacing robots to computer systems</td></tr>
<tr><td colspan="7"><u>Outline Syllabus</u>

• Different types of robots, robotic Structure, workspaces and applications
• Degrees of freedom and spatial transformations
• Orientation matrices, representation of robots in mathematical space
• Forward kinematics and inverse kinematics
• Trajectory planning and generation
• Interfacing robots
• Hands-on session on programming an industrial robot</td></tr>
</table>

<table>
<tr><td>Module Code</td><td>CS4442</td><td>Module Title</td><td colspan="4">Computer and Network Security</td></tr>
<tr><td>Credits</td><td>3</td><td rowspan="2">Hours/Week</td><td>Lectures</td><td>2</td><td rowspan="2">Pre – requisites</td><td rowspan="2">CS3032, CS3052</td></tr>
<tr><td>GPA/NGPA</td><td>GPA</td><td>Lab/Assignments</td><td>3</td></tr>
<tr><td colspan="7"><u>Learning Outcomes</u>

After completing this module, the students should be able to
• assess the security of computer systems and networks
• design secure networks and application environments</td></tr>
<tr><td colspan="7"><u>Outline Syllabus</u>

• Secure computer systems
o OS security principles, exploitations
• Secure networks
o Security devices: Firewalls, IDS, IPS, etc.
o Secure designs
• Client and server security
o Secure applications
• Secure Protocols
o Kerberos, VPN, L2TP, PPTP, IPSec, SSL, HTTPS
• Security standards</td></tr>
</table>

<table>
<tr><td>Module Code</td><td>CS4462</td><td>Module Title</td><td colspan="4">Information Security and Cryptography</td></tr>
<tr><td>Credits</td><td>3</td><td rowspan="2">Hours/Week</td><td>Lectures</td><td>2</td><td rowspan="2">Pre – requisites</td><td rowspan="2">CS3052</td></tr>
<tr><td>GPA/NGPA</td><td>GPA</td><td>Lab/Assignments</td><td>3</td></tr>
<tr><td colspan="7"><u>Learning Outcomes</u>

After completing this module, the students should be able to
• demonstrate theoretical principles of information security
• apply information security principles in information processing and exchange</td></tr>
<tr><td colspan="7"><u>Outline Syllabus</u>

• Information security models
• Access control theories and techniques
• Cryptography and related concepts, theories, standards, and methods
• Key exchange, digital certificate systems and protocols
• Application security, attacks and counter measures</td></tr>
</table>

<table>
<tr><td>Module Code</td><td>CS4482</td><td>Module Title</td><td colspan="4">Advanced Networking</td></tr>
<tr><td>Credits</td><td>3</td><td rowspan="2">Hours/Week</td><td>Lectures</td><td>2</td><td rowspan="2">Pre – requisites</td><td rowspan="2">CS3032, CS3412</td></tr>
<tr><td>GPA/NGPA</td><td>GPA</td><td>Lab/Assignments</td><td>3</td></tr>
<tr><td colspan="7"><u>Learning Outcomes</u>

After completing this module, the students should be able to
• present advance concepts and theories in networking
• present the findings after conducting research in an assigned area in networking

•</td></tr>
<tr><td colspan="7"><u>Outline Syllabus</u>

• Issues and new trends in network protocols

• Variants of OSI protocol stacks – emerging protocols

• Networking models and concepts, graph theory, mathematical and statistical models

• Highly distributed and adaptive networks

• New theories on QoS, traffic management, fault tolerance, performance and handling, etc.</td></tr>
</table>

<table>
<tr><td>Module Code</td><td>CS4492</td><td>Module Title</td><td colspan="4">Current Topics in Networking</td></tr>
<tr><td>Credits</td><td>3</td><td rowspan="2">Hours/Week</td><td>Lectures</td><td>2</td><td rowspan="2">Pre – requisites</td><td rowspan="2">CS3032, CS3412</td></tr>
<tr><td>GPA/NGPA</td><td>GPA</td><td>Lab/Assignments</td><td>3</td></tr>
<tr><td colspan="7"><u>Learning Outcomes</u>

After completing this module, the students should be able to
• describe emerging areas of networking
• critically evaluate the developmental arc of a selected emerging area of networking</td></tr>
<tr><td colspan="7"><u>Outline Syllabus</u>

• Select 3 to 6 areas of current and/or future significance in networking. For each area:
o development of the area
o theory and principles used in the area
o current status of technology
o future trends

• Conduct a comprehensive study on a selected topic in networking and prepare a report.</td></tr>
</table>

Module Code	CS4552	Module Title	Scientific Computing			
Credits	3	Hours/Week	Lectures	2	Pre – requisites	MA4052
GPA/NGPA	GPA		Lab/Assignments	3		

Learning Outcomes

After following this course, students should be able to

- analyse a computational problem and design an efficient algorithm to solve it
- optimise a scientific program under specified resource constraints
- evaluate computational methods their suitability for use in scientific and engineering problems

Outline Syllabus

- **Introduction to Scientific Computing**
- **Patterns for Parallel Programming**
  - Task decomposition, Data Decomposition, Group task, Order Tasks
  - Task Parallelism, Divide and Conquer, Geometric Decomposition, Pipelining
  - SPMD pattern, Master/Worker pattern, Fork/Join, Loop Parallelism, Shared Data, Shared Queue, Distributed Array
  - Memory Synchronisation and Fences, Barrier Synchronisation, Mutual Exclusion
- **Programming environments for scientific computing**
  - Cluster/Grid computing - MPI
  - Shared Memory Multiprocessors - OpenMP, GPU computing, CUDA programming
- **Algorithmic implementation for numerical methods**
  - Linear and non-linear systems
  - Optimisation
  - Vizualisation
  - Ordinary and partial differential equations
- **Case Studies**

- Computational Finance – Monte Carlo simulations (Geometric Brownian motion, CIR, Jump Diffusion processes)
- Data mining
- Bio-medical applications – MRI
- Computational Fluid Dynamics

<table>
<tr><td>Module Code</td><td>CS4622</td><td>Module Title</td><td colspan="4">Machine Learning</td></tr>
<tr><td>Credits</td><td>3</td><td rowspan="2">Hours/Week</td><td>Lectures</td><td>2</td><td rowspan="2">Pre – requisites</td><td rowspan="2">CS3612</td></tr>
<tr><td>GPA/NGPA</td><td>GPA</td><td>Lab/Assignments</td><td>3</td></tr>
<tr><td colspan="7"><u>Learning Outcomes</u>

After completing this module, the students should be able to
• describe the fundamental concepts in problem solving with intelligent systems
• explain the mathematical basis for the techniques used in machine learning
• evaluate and select the appropriate computational intelligence techniques for a given problem
• construct a solution for a real world application using the appropriate computational intelligence techniques</td></tr>
<tr><td colspan="7"><u>Outline Syllabus</u>

• Problem Solving With Computational Intelligence: The Basic Concepts
• Single-Layer and Multi-Layer Feed-forward Neural Networks
• Feedback Neural Networks
• Associative Memories
• Self-Organizing Feature Maps
• Support Vector Machines
• Genetic Algorithms
• Fuzzy Sets and Fuzzy Logic
• Fuzzy Neural Networks
• Swarm Intelligence and Ant Colony Optimization, Effectiveness and Computational Complexity of Computational Intelligence Models.</td></tr>
</table>

Module Code	MA4052	Module Title	Numerical Analysis for Scientific Computing			
Credits	2	Hours/Week	Lectures	2	Pre – requisites	MA1032
GPA/NGPA	GPA		Lab/Assignments	0		

Learning Outcomes

After completing this module, the students should be able to

- understand a spectrum of advanced concepts in numerical analysis
- apply them to solve engineering and scientific problems

Outline Syllabus

- **Modeling, Computers, and Error Analysis**
  - Mathematical Modeling and Engineering Problem Solving
  - Programming and Software
  - Approximations and Round-Off Errors
  - Truncation Errors and the Taylor Series
- **Solution of Linear Algebraic Equations**
  - Gauss- Jordan Elimination and Backsubstitution
  - LU- Decomposition and its Applications
  - Tri- Diagonal and Band-Diagonal Systems of Equations
  - Singular Value Decomposition, Cholesky Decomposition & QR-Decomposition
  - Spare Liner Systems
- **Eigensystems**
  - Jacobi Transformations
  - Real Symmetric Matrices
  - Reductions to Tri- Diagonal From (Givens & Householder methods)
  - Eigenvalues & Eigenvectors of Tri-Diagonal Matrices
  - Hermitian Matrices
- **Modelling of data**

- Least Squares as a Maximum Likelihood Estimator
- Non- linear Models
- Robust Estimation
- Markov Chain Monte Carlo
- Gaussian Process Regression

- **Integration of Ordinary Differential Equations**
  - Runge – Kutta Method
  - Stiffness and Multistep Method
  - Richardson Extrapolation
  - Second Order Conservative Equations
  - Multi – step, Multivalue and Predictor-Corrector Methods
- **Two- Point Boundary Value Problems**
  - Shooting Method
  - Relaxation Methods
  - Automated Allocation of Mesh Points
  - Handling Internal Boundary Conditions or Singular Points
- **Partial Differential Equations**
  - Flux- Conservative Initial value Problems
  - Diffusive Initial Value Problems
  - Initial Value Problems in multi-Dimensions
  - Fourier & Cyclic Reduction Methods, Relaxation Methods and Multi-grid Methods for Boundary Value Problems

<table>
<tr><td>Module Code</td><td>CS2812</td><td>Module Title</td><td colspan="4">Visual Programming</td></tr>
<tr><td>Credits</td><td>2</td><td rowspan="2">Hours/Week</td><td>Lectures</td><td>1</td><td rowspan="2">Pre – requisites</td><td rowspan="2">None</td></tr>
<tr><td>GPA/NGPA</td><td>GPA</td><td>Lab/Assignments</td><td>3</td></tr>
<tr><td colspan="7"><u>Learning Outcomes</u>

After completing this module, the students should be able to
• Develop a working program for specified programming problem using a visual programming environment</td></tr>
<tr><td colspan="7"><u>Outline Syllabus</u>
• Introduction to the concept of visual programming
• Introduction to visual programming environments
• Practice of visual programming using .NET Framework
o Objects, Properties, Events and Methods
o Variables, Data Types and Controls
o Use of Forms and Controls to create User Interfaces
o Program Control Flow
o String and file manipulation
o Arrays
o Procedures and Functions
o Exception Handling
o Database Programming</td></tr>
</table>

Module Code	CS2842	Module Title	Computer Systems			
Credits	2	Hours/Week	Lectures	2	Pre – requisites	None
GPA/NGPA	GPA		Lab/Assignments	0		

Learning Outcomes

After completing this module, the students should be able to

- understand the representations used for numbers and text, computer arithmetic, the functions of the components of a CPU, how main memory is organised, the architecture of a microprocessor, models for input/output
- interface external devices to the microprocessors
- develop assembly programs for microprocessors

Outline Syllabus

- **Introduction**
  - Levels of abstraction
  - Instruction set level
  - Hardware design level
  - Role of the computer architect.

- **Review of Data representation**
- **CPU organisation and operation**
- **Memory and storage organisation**
- **Introduction to Operating Systems**
- **I/O and Standard interfaces**
- **Introduction to assembly programming**

Module Code	CS 2832	Module Title	Modular Software Development			
Credits	2.5	Hours/Week	Lectures	2	Pre – requisites	None
GPA/NGPA	GPA		Lab/Assignments	3/2		

Learning Outcomes

After completing this module, the students should be able to

- use object oriented programming concepts to solve programming problems
- develop software packages using object oriented programming methodologies
- describe principles, concepts and practice of software engineering
- explain the methods and processes of constructing different software systems
- apply techniques and tools of software engineering within the context of systematic construction of quality software

Outline Syllabus

- **Introduction to Object-Oriented Programming Languages**
    - Features of a OOP language, data types, variables and literals, programming constructs
- **OOP Concepts**
    - Classes, public/private declarations, class access, pointers, inheritance, polymorphism, interfaces and packages, exception handling, object serialization, memory management
- **GUIs**
    - Layout managers, GUI components, event handlers
- **The Software Process**
    - Software development life cycle, software process models, software quality metrics, project scheduling
- **Requirements Engineering**
    - Requirements elicitation, requirements analysis and specifications, formal specification techniques
- **Software Designing**
    - Data design, software architecture, functional independence, interface design, component level design, coding standards
- **Software Testing**

- Verification and validation, testing plan, testing technique, white/black box testing, testing case design, debugging practices