



**Fergusson College (Autonomous)  
Pune**

**Learning Outcomes-Based Curriculum  
For  
M.Sc. I - Computer Science**

**With effect from June 2019**

### Programme Structure

Year	Course Code	Course Title	Credits
First Year (Semester - I)	CSC4101	Data Mining and Data Warehousing	4
	CSC4102	Analysis of Algorithms and Computing	4
	CSC4103	Operating System Internals	4
	CSC4104	Python Programming	4
	CSC4105	Computer Science Practical - I	4
	CSC4106	Project - I	4
First Year (Semester - II)	CSC4201	Programming in C#	4
	CSC4202	Full Stack-I: Web UI and Responsive UI Framework	4
	CSC4203	Business Intelligence and Analytics <b>OR</b>	4
	CSC4204	Robotics and Artificial Intelligence <b>OR</b>	
	CSC4205	MOOC-I <b>OR</b>	
	CSC4206	Principles of Programming Language (General Elective - I)	4
	CSC4207	Networking and Information Security <b>OR</b>	
	CSC4208	Software Quality Assurance and Testing <b>OR</b>	
	CSC4209	MOOC-II <b>OR</b>	
	CSC4210	Parallel Computing (General Elective - II)	
	CSC4211	Computer Science Practical - II	4
	CSC4212	Project - II	4
Second Year (Semester - III)	CSC4301	Soft Computing	4
	CSC4304	Full Stack-II : MEAN stack	4
	CSC4304	Software Project Management	4
	CSC4305	Cloud Computing	4
		Machine Learning	
		MOOC-III	
	CSC4306	Modelling and Simulation (General Elective - III)	4
		Computer security and Cyber Forensics	
		Digital Image Processing	
		Data science using R	
CSC4306	MOOC-IV	4	
	Natural Language Processing (General Elective - IV)		
	Project -1 group - 1hr / week		
CSC4308	Project -1 group - 1hr / week	4	
Second Year (Semester - IV)	CSC4401	Industrial Training	8
<b>Total Credits</b>			<b>80</b>

## Programme learning outcomes relating to M.Sc. in Computer Science

PO1	Specialist knowledge of computer science theories, methods, practices and strategy
PO2	Understanding of a range of programming languages
PO3	Understanding of computing architecture, construction and design
PO4	Understanding of computing software, tools, packages and design
PO5	Awareness of and ability to cope with and adapt to rapid technological changes
PO6	Teamwork and leadership skills
PO7	Analytical and critical thinking skills
PO8	Creative thinking skills
PO9	Time management and organization skills
PO10	Written and oral communication skills, including presentations and report writing
PO11	Strong problem-solving, analysis and decision-making abilities
PO12	Equip with knowledge and skills necessary to support for their career in Information Security, Net Security and Cryptography

### Course Learning Outcomes (Course / paper wise)

- Learn the Algorithm Analysis techniques and use of data structures in improving algorithm performance
- To learn new emerging technology/framework.
- To develop Basic and advanced concepts of algorithms for data mining.
- To implement analytical and predictive modeling applications.
- Get knowledge of cloud OS and architectures
- Enable students to get sound understanding of Advanced Networking Concepts
- Understand working of internals of Unix operating system and implement Unix system calls
- Develop the skills to build accessible, usable, attractive and responsive User Interfaces for web applications
- Propose the use of certain technologies by implementing them in the C # programming language to solve the given problem
- Understand the advanced concepts of computer networking and write socket program for client server communication.
- Implement Artificial Intelligence techniques to develop intelligent system
- To develop new and imaginative solutions to security problems under study.
- Learn to select and apply project management techniques for process modelling, planning, estimation, process metrics and risk management.
- Analyse the data and identify trends using machine learning.

**CSC4101 - Data Mining and Data Warehousing**  
**Credits: 4**

<b>Learning Outcomes</b>	<b>Suggested Pedagogical Processes</b>
Student will understand Data processing and data quality.	Lecture method with use of ICT
The student will get knowledge of Modeling and design of data warehouses	Lecture method with use of ICT
To develop Basic and advanced concepts of algorithms for data mining	Case studies and assignments method
To understand Data mining tool and practical experience of applying data mining algorithms	Experiential learning through practical problem Solving and assignment

<b>Unit No.</b>	<b>Title of Unit and Contents</b>
I	<b>Introduction to Data Mining</b> Basic Data Mining Tasks, DM versus Knowledge Discovery in Databases, Data Mining Issues, Data Mining Metrics, Social Implications of Data Mining, Overview of Applications of Data Mining
II	<b>Introduction to Data Warehousing</b> The Need for Data Warehousing, Operational V/s Decisional Support System; Benefits of Data Warehousing ;Features of a Data Warehouse; Meta Data, ETL Process. Data Warehouse Architecture, Different Types of Architecture; Data Warehouse and Data Marts; Data Warehousing Design Strategies. OLAP and Data Cubes. Dimensional Data Modelling-star, snowflake schemas, Data Pre-processing - Need, Data Cleaning, Data Integration & Transformation, Data Reduction,
III	<b>Data Pre-processing</b> Why Pre-processing? Data Cleaning; Data Integration; Data Reduction: Attribute subset selection, Histograms, Clustering and Sampling; Data Transformation & Data Discretization: Normalization, Binning, Histogram Analysis and Concept hierarchy generation.
IV	<b>Data Mining Techniques</b> Frequent item-sets and Association rule mining: A priori algorithm, Use of sampling for frequent item-set, FP tree algorithm, Graph Mining: Frequent sub-graph mining, Tree mining, Sequence Mining
V	<b>Classification &amp; Prediction</b> Decision tree learning, Construction, performance, attribute selection, Issues: Over-fitting, tree pruning methods, missing values, continuous classes, Classification and Regression Trees (CART), Bayesian Classification, Bayes Theorem, Naïve Bayes classifier, Inference, Parameter and structure learning, Linear classifiers, Prediction, Linear regression, Non-linear regression
VI	<b>Clustering</b> k-means, Expectation Maximization (EM) algorithm, Hierarchical clustering, Correlation clustering
VII	<b>Accuracy Measures</b> Precision, recall, F-measure, confusion matrix, cross-validation, bootstrap <b>Data Exploration</b>

	Types of Attributes Statistical Description of Data Data Visualization Measuring similarity and dissimilarity
VIII	<b>Brief overview of advanced techniques</b> Text mining, Web Mining

### Learning Resources

1. Han, Data Mining: Concepts and Techniques, Elsevier
2. Margaret H. Dunham, S. Sridhar, Data Mining – Introductory and Advanced Topics, Pearson Education
3. Tom Mitchell, - Machine Learning, McGraw-Hill, 1997
4. R.O. Duda, P.E. Hart, D.G. Stork. Pattern Classification. Second edition. John Wiley and Sons, 2000.
5. Christopher M. Bishop, - Pattern Recognition and Machine Learning, Springer 2006
6. Raghu Ramkrishnan, Johannes Gehrke, Database Management Systems, Second Edition, McGraw Hill International
7. Ian H.Witten, Eibe Frank Data Mining: Practical Machine Learning Tools and Techniques, Elsevier / (Morgan Kauffman)

[Research-Papers]: Some of the relevant research papers that contain recent results and developments in data mining field

**CSC4102 - Analysis of Algorithms and Computing**  
**Credits: 4**

Learning Outcomes	Suggested Pedagogical Processes
To provide foundation in algorithm design and analysis	Lecture method with use of ICT, algorithm assignment
To develop an ability and understanding of designing algorithms in context of space and time complexity	Lecture method and assignment
To understand different design strategies	Problem solving method
To understand classical problem and solutions	Lecture method with use of ICT
To understand the use of data structures in improving algorithm performance	Programming assignment

Unit No.	Title of Unit and Contents
I	<b>Introduction</b> The Role of Algorithms in Computing Algorithm as a technology Fundamentals of Algorithmic Problem Solving The Analysis Framework, Asymptotic Notations Mathematical Analysis of Non-recursive & Recursive Algorithms Linear and Nonlinear sorting
II	<b>Design Strategies</b> Decrease and Conquer: Types, Insertion Sort, Topological Sorting Divide and Conquer: Concept, Sorting algorithms, Strassen's matrix multiplication
III	<b>Advanced Design and Analysis Techniques</b> Greedy Method: Types, Knapsack problem, Spanning Trees, An activity-selection problem, Optimal Merge Pattern, Huffman codes Dynamic Programming :Concept, Matrix Chain Multiplication, Longest Common Subsequence, Shortest Path, ,Traveling Sales Person's (TSP) problem, 0/1 knapsack problem Backtracking : Concept, Graph coloring and Hamiltonian cycle Branch and Bound : Concept, LCBB, TSP, 0/1 knapsack problem
IV	<b>Problem classification</b> Nondeterministic algorithm The class of P, NP, NP-hard and NP - Complete problems Cook's theorem

**Learning Resources**

1. Ellis Horowitz, Sartaj Sahni & Sanguthevar Rajasekaran, Computer Algorithms, Galgotia.
2. T. Cormen, C. Leiserson, & R. Rivest, Algorithms, MIT Press, 1990
3. A. Aho, J. Hopcroft, & J. Ullman, The Design and Analysis of Computer Algorithms, Addison Wesley, 1974
4. Donald Knuth, The Art of Computer Programming (3 vols., various editions, 1973-81), Addison Wesley

**CSC4103 - Operating System Internals**  
**Credits: 4**

<b>Learning Outcomes</b>	<b>Suggested Pedagogical Processes</b>
Students will be able to understand architecture of UNIX operating system.	The Lecture method with use of ICT
They will understand all kernel data structures.	The Lecture method with use of ICT
Students will get practical knowledge of how kernel manages signals and memory of the system.	Practical demonstration Method
Students will be able to implement different file concepts using C programs.	Practical demonstration program assignment
Students will be able to implement different Unix system calls using „C.“	Practical demonstration program assignment

<b>Unit No.</b>	<b>Title of Unit and Contents</b>
I	<b>General Overview</b> System Structure, User Perspective, Assumptions about Hardware, Architecture of UNIX Operating System. Concepts of Linux Programming- Files and the File system, Processes, Users and Groups, Permissions, Signals, Inter-process Communication
II	<b>Introduction to the kernel</b> Architecture of the UNIX operating system, an overview of the file subsystem, Kernel data structures for file, Concept of program, process, Kernel data structures for process
III	<b>The buffer cache</b> Need of buffer cache, Buffer headers, Structure of the buffer pool, Scenarios of retrieval of a buffer, algorithm getblk, Race condition for free buffer and race for locked buffer in getblk, Algorithm for releasing a buffer – brelse, Reading and writing disk blocks, algorithms bread, breada and bwrite
IV	<b>Internal representation of file</b> Inodes - disk inode, in-core inode, Accessing and releasing inode, algorithm iget and iput, Structure of a regular file, Conversion of byte offset to block number, Directories, Conversion of a pathname to an inode, Super block, Inode assignment to a new file, algorithm ialloc, Freeing inode, algorithm ifree, Allocation of disk blocks, algorithm alloc

V	<p><b>File and Directory I/O</b>  open, read, write,lseek, closing of file using close, Creation of regular file, special files, pipes, System call stat, fstat, dup, dup2, Mounting of file system, atomic operations, dup2, sync, fsync, and fdatsync, fcntl, /dev/fd, stat, fstat, lstat, file types, fdatsync, fcntl, /dev/fd, stat, fstat, lstat, file types, Set-User-ID and Set-Group-ID, file access permissions, ownership of new files and directories, access function, umask function, chmod and fchmod, sticky bit, chown, fchown, and lchown, file size, file truncation, file systems,link, unlink, remove, and rename functions, symboliclinks, symlink and readlink functions, file times, utime,mkdir and rmdir, reading directories, chdir, fchdir, and getcwd, device special files Scatter/Gather I/O, Mapping Files into Memory, Advice for Normal File I/O, I/O Schedulers and I/O Performance, Directories, Copying and Moving files, Device Nodes, Out-of-Band Communication</p>
VI	<p><b>Process Environment, Process Control and Process Relationships</b>  Process states and transitions, layout of system memory, the context of a process, saving the context of a process, sleep, process creation, signals, process termination, awaiting process termination, invoking other programs, the user id of a process, changing the size of the process, The Shell, Process Scheduling.  Process termination, environment list, memory layout of C program, shared libraries, environment variables, setjmp and longjmp, getrlimit and setrlimit, process identifiers,fork, vfork, exit, wait and waitpid, waitid, wait3 and wait4, race conditions, exec, changing user IDs and group IDs,system function, user identification, process times.  The Process ID, Running a New Process, Terminating a Process, Waiting for Terminated Child Processes, Users and Groups, Daemons, Process Scheduling, Yielding the Processor, Process Priorities, Processor Affinity.</p>
VII	<p><b>Memory Management</b>  The Process Address Space, Allocating Dynamic Memory, Managing Data Segment, Anonymous Memory Mappings, Advanced Memory Allocation, Debugging Memory Allocations, Stack-Based Allocations, Choosing a Memory Allocation Mechanism, Manipulating Memory, Locking Memory, Opportunistic Allocation Swapping, Demand Paging, Hybrid system with swapping and demand paging</p>
VIII	<p><b>Signal Management</b>  concepts,signal function, unreliable function interrupted system calls, re-entrant I/O semantics, reliable-signal technology, kill and raise, alarm and pause, signal mask, sigpending, sigsetjmp and siglongjmp, sigsuspend, abort, system function sigset, Basic Signal Management, Sending a Signal, Reentrancy, Signal Sets, Blocking and Non-blocking Signal Management, Sending a Signal with a Payload</p>

### Learning Resources

1. Richard Stevens, Advanced Programming in the UNIX Environment, Addison-Wesley
2. Maurice J. Bach, The Design of the UNIX Operating System, PHI
3. Robert Love, Linux System Programming, O'Reilly



**CSC4104 - Python Programming**  
**Credits: 4**

Learning Outcomes	Suggested Pedagogical Processes
To understand importance of Python as a scripting language.	The Lecture method with use of ICT, practical demonstration.
Develop a skill to implement Python Programming.	Solving practical assignments with small application development.
Should be able to use standard programming constructs like arrays, lists, tuples.	
Knowledge of the modules and packages will help in writing reusable code.	
Knowledge of regular expression will help students in pattern matching and searching, also in validating email addresses and urls.	
Studying database concepts will help in writing complete application where data is stored in database.	
Introduction to Web crawler and Image Processing with Google Vision API will give exposure to current technologies.	

Unit No.	Title of Unit and Contents
I	<b>Introduction To Python</b> Introduction Various IDEs
II	<b>Data Types:</b> Numeric data types: int, float, complex string list and list slicing Tuple
III	<b>Control Flow, Functions, Modules And Packages</b> Control Flow Conditional blocks using if, else and elif Simple for and while loops in python For loop using ranges, string, list and dictionaries Loop manipulation using pass, continue, break and else Functions Arguments, Lambda Expressions, Function Annotations Modules Organizing python projects into modules Importing own module as well as external modules Packages Programming using functions, modules and external packages
IV	<b>Data Structures</b>

	Lists as Stacks, Queues, Comprehensions Tuples and sequences Sets Dictionaries
V	<b>Python File Operation</b> Reading config files in python Writing log files in python Understanding read functions, read(), readline() and readlines() Understanding write functions, write() and writelines() Manipulating file pointer using seek Programming using file operations
VI	<b>Object Oriented Programming</b> Concept of class, object and instances Constructor, class attributes and destructors, Inheritance, overlapping and overloading operators, Adding and retrieving dynamic attributes of classes Programming using Oops support
VII	<b>Regular Expression</b> Powerful pattern matching and searching Power of pattern searching using regex Real time parsing of networking or system data using regex Password, email, url validation using regular expression Pattern finding programs using regular expression
VIII	<b>Database Interaction SQL</b> Database connection using python Creating and searching tables Reading and storing config information on database Programming using database connections
IX	<b>Image Processing with Google Vision API</b> Google's Vision API offerings of Vision API Vision API Client Library for Python
X	<b>Writing Web Crawler using Python</b>

### Learning Resources

1. Learning Python, O'Really
2. Programming Python, O'Really
3. <https://docs.python.org/3/tutorial/>
4. <https://www.datacamp.com/community/tutorials/beginner-guide-google-vision-api>

**CSC4105 - Computer Science Practical - I**  
**Credits: 4**

<b>Learning Outcomes</b>	<b>Suggested Pedagogical Processes</b>
After completing assignments students will learn Hands-on Python experience for professional advancement	

<b>Sr. No.</b>	<b>Title of Experiment / Practical</b>
1	Data Types: String and List
2	Python Functions
3	Data Structures: Stacks, Queues, Tuples, sets, Dictionaries
4	Data Structures: Stacks, Queues, Tuples, sets, Dictionaries
5	File Handling
6	File Handling
7	Regular Expression
8	Object Oriented Programming: Inheritance, Abstract classes and interfaces
9	Object Oriented Programming: Inheritance, Abstract classes and interfaces
10	Working with Database
11	Image Processing with Google Vision API
12	Web Scrapping

**CSC4106 - Project - I**  
**Credits: 4**

**Objective:**

**The objective of project is to make the students understand Requirement analysis, design and implementation cycle. Any open problem statement can be taken for implementation. The system can be designed in any programming language implemented in any platform.**

The Project can be platform, Language and technology independent. Project will be evaluated by project guide. Assessment will be done weekly in the respective batch. Evaluation will be on the basis of weekly progress of project work, progress report, oral, results and documentation and demonstration.

You should fill your status of the project work on the progress report and get the Signature of project guide regularly. Progress report should sharply focus how much time you have spent on specific task. (The format of progress report is given as follow.) You should keep all signed progress report. Project will not be accepted if progress report is not submitted and all responsibility remains with student.

**Project Progress Report**

<b>Roll No and Name of the Student</b>	
<b>Title of the project</b>	
<b>Project guide name</b>	

<b>Sr. No</b>	<b>From Date</b>	<b>To Date</b>	<b>Details of Project work</b>	<b>Project guide sign (with date)</b>

**Project Guide**

**CSC4201 - Programming in C#**  
**Credits: 4**

<b>Learning Outcomes</b>	<b>Suggested Pedagogical Processes</b>
After completion of the course the student will be able to learn powerful Object-Oriented principles applied in C#.	Use of ICT for effective learning.
Able to use the C# programming language for developing various software applications.	Whenever possible explanation of theoretical concept can be followed by a live demo in class.
Students will be able to propose the use of certain technologies by implementing them in the C # programming language to solve the given problem.	Home assignments for learning through experiments.
Will be able to write form based application catering different needs of scenarios.	Teaching Through Conversation i.e. encouraging students to think of different project scenarios where a concept / technique can be used.
Knowing concepts like multithreading will help to write multi threaded application.	Solving practical assignments with small application development.
Familiarity of Entity Framework will strengthen their knowledge of Projects in terms of identifying entities of projects, mapping to database tables.	
Exposure to LINQ will help implementing optimized querying techniques.	
Introduction to WCF will open pool of techniques for communication over the net.	

<b>Unit No.</b>	<b>Title of Unit and Contents</b>
I	<b>DOTNET Framework</b> Introduction to DOTNET DOT NET class framework Common Language Runtime Overview, Elements of .NET application, Memory Management, Garbage Collection Common Language Integration Common type system Assemblies: Private assemblies, Shared assemblies
II	<b>Core C#</b> Language features: Variables and Expressions, type conversion, String and String Builder, Flow Control, Functions, Passing Parameters by value and by ref, Anonymous Types, Debugging and error handling, exception handling ( System Defined and User Defined) Object Oriented Concepts

	<p>Defining classes, class members, Interfaces, Properties, Access modifiers, properties and auto-implemented properties.</p> <p>Inheritance – Virtual Methods, polymorphism, hiding Methods, Abstract classes and methods, Sealed classes and methods, is and as Operators</p> <p>Collections, Comparisons and Conversions</p> <p>Defining and using collections, Indexers, iterators, Type comparison, Value Comparison, Overloading Conversion operators</p> <p>Generics</p> <p>Using generics, Defining Generics, generic Interfaces, Generic methods, Generic Delegate</p>
III	<p><b>Delegates , lambdas and events</b></p> <p>Simple delegate, Multicast delegate, Action &lt;T&gt; and Func&lt;T&gt; delegate, Anonymous methods, Lambda Expressions</p> <p>Events: Event Publisher, Event Listener</p>
IV	<p><b>Window Programming</b></p> <p>Window Controls:</p> <p>Common Controls, Container Controls (Group box and Tab controls), Menus and Toolbars, Dialogs</p> <p>Deploying Window Application</p> <p>Click Once deployment, Introduction to WIX setup</p>
V	<p><b>Data Access</b></p> <p>File System Data</p> <p>XML</p> <p>Databases and Entity Framework</p> <p>Data Binding</p>
VI	<p><b>LINQ</b></p> <p>Overview, Standard Query Operators, LINQ with SQL, LINQ with Xml, LINQ with Collection</p>
VII	<p><b>Reflection, Metadata and Dynamic Programming</b></p> <p>Standard Attributes and Custom Attributes, System.Type class, Metadata, Reflection, Assembly class, Memberinfo class, Dynamic Type, dynamic vs var, Dynamic Language Runtime (DLR)</p>
VIII	<p><b>Multithreading</b></p> <p>Thread life cycle, Thread synchronization, Understanding Thread Pool, Background worker, introduction to Task, Async programming</p>
IX	<p><b>Introduction to Windows Communication Foundation</b></p> <p>WCF Architecture, WCF Application Components, WCF simple Programming, Data contracts, message contracts</p> <p>end points, type of binding one way, two way, Hosting wcf in windows service, IIS. Introduction to REST using WCF</p>

## Learning Resources

1. Beginning Visual C#, Wrox Publication
2. Professional Visual C#, Wrox Publication
3. <https://docs.microsoft.com/en-us/dotnet/csharp/>

**CSC4202 - Full Stack - I: Web UI and Responsive UI Framework**  
**Credits: 4**

<b>Learning Outcomes</b>	<b>Suggested Pedagogical Processes</b>
Develop the skills to build accessible, usable and attractive User Interfaces for your web applications.	The Lecture method with use of ICT, practical demonstration
To make the user interaction as very simple and most efficient;	Practical Assignments with small application development.

<b>Unit No.</b>	<b>Title of Unit and Contents</b>
I	<p><b>Overview of HTML5</b>            History, Vision &amp; Future of HTML5, Structure of a Web Page: HTML5 DOCTYPE, Page Encoding, HTML5 Mark-up: New And Updated Elements, Structural Elements, New Attributes, Deprecated Elements And Attributes; HTML5 And CSS3, Browser Support, Forms, Audio and Video, Canvas, SVG, Local Storage, Geo location.</p>
II	<p><b>Introducing CSS3</b>            History of CSS, Browser Support, HTML5, Selectors and Pseudo Classes, Fonts and Text Effects, Colours, Gradients, Background Images, and Masks, Borders and Box Effects, Transitions, Transforms, and Animations, Layout: Columns and Flexible Box, Vendor Prefixes, Embedding Media.</p>
III	<p><b>Overview of JavaScript</b>            Introduction to Javascript, Javascript Statements, Javascript Keywords, Javascript Functions, Javascript Programs, Javascript Operators, Function Parameters, Function Return Values, Javascript Data Types, Primitive Types,            Working with Objects : Object Overview, Object creation            Adding Properties to Objects, Adding Methods to Objects,            Javascript Conditional Statements, Javascript Loops &amp; Iteration, Enumerating properties, Callbacks, JSON</p>
IV	<p><b>Bootstrap</b>            Introduction, Grid, Components, Plugins.</p>
V	<p><b>jQuery and AJAX</b>            Introduction, Attributes, Elements, Selectors, jQuery Object, Traversing, CSS manipulation, Data Methods, Utility Methods, Events, Effects, Key Concepts of Ajax, jQuery's Ajax-Related Methods, Ajax and Forms, Working with JSONP, Ajax Events.</p>



## Learning Resources

1. Beginning HTML5 and CSS3 By Christopher Murphy, Divya Manian, Oliver Studholme and Richard W.Clark (APress)
2. Beginning JavaScript By Jeremy McPeak and Paul Wilton(Wrox)
3. Head First HTML5 Programming, Building Web Apps with JavaScript By Eric Freeman, Elisabeth Robson (O'Reilly).
4. Beginning JQuery By Jack Franklin (APress)
5. Bootstrap By Jake Spurlock (O'Reilly)
6. Head First Ajax By Rebecca M. Riordan (O'Reilly)

### Important URLs:

1. <https://www.w3schools.com>
2. <http://getbootstrap.com>

**CSC4203 - Business Intelligence and Analytics**  
**Credits: 4**

<b>Learning Outcomes</b>	<b>Suggested Pedagogical Processes</b>
To implement analytical and predictive modeling applications.	Lecture method with use of ICT
Apply BI to solve practical problems.	Lecture method with use of ICT
Analyze the problem domain, use the data collected in enterprise. Apply the appropriate data mining technique, interpret and visualize the results and provide decision support.	Case studies and assignments method
To provide knowledge on how to gather and analyze large sets of data to gain useful business understanding.	Experiential learning through practical problem Solving and assignment

<b>Unit No.</b>	<b>Title of Unit and Contents</b>
I	<b>Introduction to Business Intelligence</b> Definition and History of BI, Transaction processing versus analytical processing, BI implementation , Major tools and techniques of BI
II	<b>Data warehousing</b> Definition and concepts, Data warehouse architecture, ETL process, data warehouse development, ,Comparison of OLTP and OLAP Top down vs. Bottom up, Data Mart vs. EDW, Implementation issues, Real-time data warehousing, Slicing, dicing and cross applications reporting and complex data analysis
III	<b>Business performance management</b> Key performance indicators and operational metrics, Balanced scorecard , Six Sigma, Dashboards and scorecards
IV	<b>Introduction to Business Analytics</b> Understanding Business Intelligence and Analytics Data analytic lifecycle, various phases of Data analytic lifecycle, business analytics, Business decision, Features of Business analytics, Types of business analytics.

V	<b>Basic Statistics in Business Analytics</b> Introduction to Probability, Probability Distributions, Connection with Statistical Distributions, Statistical Properties (Mean, Mode, Median, Moments, Standard Deviation, etc.), Common Probability Distributions (Discrete, Binomial, Normal), Other Probability Distributions (Chi-Square, Poisson), Joint and Conditional Probabilities
VI	<b>Basics of Modeling</b> Differentiating descriptive, predictive, and prescriptive analytics, Data mining vs data analytics, Industrial problem solving process, Decision needs and analytics, stakeholders and analytics, SWOT analysis
VII	<b>Data Visualization Techniques</b> Data Preparation and Exploration, Importance of data quality, Dealing with missing or incomplete data , Data Classification
VIII	<b>Modeling Techniques</b> Introduction to Common Modeling Techniques Cluster Analysis (Unsupervised Learning) Classification & Prediction (Supervised Learning) Classification - Training & Testing Sampling Data in Classification
IX	<b>Predictive analytics</b> Predictive modeling and Analysis - Regression Analysis, Multi-collinearity, Correlation analysis, Rank correlation coefficient, Multiple correlation, Least square, Curve fitting

### Learning Resources:

1. EfraimTurban, Ramesh Sharda, Dursun Delen, and David King , Business Intelligence: A Managerial Approach, 2nd Edition, PEARSON 2012, ISBN-10: 0-13-610066-X, ISBN- 13: 978-0-13-610066-9
2. Simon Miller and William Hutchinson , Oracle Business Intelligence Applications, McGraw Hill Education 2013, ISBN-10: 93-5134-153-4,ISBN-13: 978-93-5134-153-6
3. Gert H.N. Laursen, JesperThorlund , Business Analytics for Managers: Taking Business Intelligence beyond Reporting Paperback , 2013
4. .Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today"s Business by AmbigaDhiraj, Wiely CIO Series.
5. Data Science & Big Data Analytics" by David Dietrich, Barry Hiller, , EMC education services, Wiley publications, 2012
6. "Business analytics: the next frontier for decision sciences." By Evans , James R., and Carl H. Lindner, Decision Line
7. Eric Siegel, Thomas H. Davenport, "Predictive Analytics: The Power to Predict Who Will Click, Buy, Lie, or Die", Wiley, 2013

**CSC4204 - Robotics and Artificial Intelligence**  
**Credits: 4**

<b>Learning Outcomes</b>	<b>Suggested Pedagogical Processes</b>
To present an overview of artificial intelligence (AI) principles and approaches	Lecture method with use of ICT, problem assignment
To develop a basic understanding of the building blocks of AI	Lecture method and problem solving
To understand basic of robotics	Lecture method and assignment
To apply AI techniques to different real word problems and games.	Class room assignments, brain storming, participative learning

<b>Unit No.</b>	<b>Title of Unit and Contents</b>
I	<b>Introduction to Artificial Intelligence</b> What is AI? Early work in AI AI and related fields AI problems and Techniques
II	<b>Problems, Problem Spaces and Search</b> Defining AI problems as a State Space Search: examples Production Systems Search and Control Strategies Problem Characteristics Issues in Design of Search Programs Additional Problems
III	<b>Heuristic Search Techniques</b> Generate-and-test Hill Climbing Best First Search Problem Reduction Constraint Satisfaction Mean-Ends Analysis
IV	<b>Knowledge Representation</b> Representations and Mappings Approaches to Knowledge Representation Knowledge representation method Propositional Logic Predicate logic Representing Simple facts in Logic Representing Instances and Isa relationships Computable Functions and Predicates Resolution Forward and backward chaining
V	<b>Slot – and – Filler Structures</b> Weak Structures Semantic Networks Frames

	Strong Structures Conceptual Dependencies
VI	<b>Game Playing</b> Minimax Search Procedures Adding alpha-beta cutoffs Uncertainty Reasoning: Basic Probability Axioms, Baye's Rule, Bayesian Classification, Certainty Factor Theory, Dempster Shafar Theory.
VII	<b>Learning</b> What is learning? Rote Learning Learning by taking advice Learning in problem solving Learning from examples Explanation based learning
VIII	<b>Robotics</b> What is a Robot? Definition, History of Robots Types of Robot Control. Robot Components , Architectures

### Learning Resources

1. Artificial Intelligence, Tata McGraw Hill, 2nd Edition, by Elaine Rich and Kevin Knight.
2. Introduction to Artificial Intelligence and Expert System, Prentice Hall of India Pvt. Ltd., New Delhi, 1997, 2nd Printing, by Dan Patterson.

**CSC4206: Principles of Programming Language**  
**Credits: 4**

<b>Learning Outcomes</b>	<b>Suggested Pedagogical Processes</b>
Student will understand language processing and the base quality structure of languages.	Lecture method with use of ICT
The student will get knowledge of understanding the behavior of languages.	Lecture method with use of ICT
To develop Basic utilities of language and functionality modules.	Case studies and assignments method
To understand the practical experience of applying programming languages	Experiential learning through practical problem Solving and assignment

	<b>Title and Contents</b>
<b>Unit -I</b>	<p><b>Introduction</b></p> <p>The Art of Language Design The Programming Language Spectrum Why Study Programming Languages? Compilation and Interpretation Programming Environments</p>
<b>Unit -II</b>	<p><b>Non-Imperative Programming Models: Functional, Logic Languages</b></p> <p>Common LISP Basic LISP Primitives ( FIRST, REST, SETF, CONS, APPEND, LIST,NTHCDR, BUTLAST, LAST, LENGTH, REVERSE, ASSOC) Procedure definition and binding, DEFUN, LET Predicates and Conditional, EQUAL, EQ, EQL, =, MEMBER, LISTP, ATOM, NUMBERP, SYMBOLP, NIL, NULL, IF, WHEN, UNLESS, COND, CASE Procedure Abstraction and Recursion Turbo Prolog Introduction, facts, Objects and Predicates, Variables, Using Rules, Controlling execution fail and cut predicates</p>
<b>Unit -III</b>	<p><b>Names, Scopes, and Bindings</b></p> <p>The Notion of Binding Time Object Lifetime and Storage Management: Static Allocation, Stack-Based Allocation, Heap-Based Allocation, Garbage Collection Scope Rules Static Scoping, Nested Subroutines, Declaration Order, Dynamic Scoping The meaning of Names in a Scope Aliases, Overloading, Polymorphism and Related Concepts The Binding of Referencing Environments Subroutine Closures, First-Class Values and Unlimited Extent,</p>

	<p>Object Closures</p> <p>Macro Expansion</p>
<b>Unit - IV</b>	<p><b>Control Flow</b></p> <p>Expression Evaluation</p> <p>Precedence and Associativity, Assignments, Initialization, Ordering Within Expressions, Short-Circuit Evaluation</p> <p>Structured and Unstructured Flow</p> <p>Structured Alternatives to goto Sequencing</p> <p>Selection</p> <p>Short-Circuited Conditions, Case/Switch Statements</p> <p>Iteration</p> <p>Enumeration-Controlled Loops, Combination Loops, Iterators, Logically Controlled Loops</p> <p>Recursion</p> <p>Iteration and Recursion, Applicative and Normal -Order Evaluation</p>
<b>Unit - V</b>	<p><b>Data Types</b></p> <p>Introduction</p> <p>Primitive Data Types</p> <p>Numeric Types</p> <p>Integer</p> <p>Floating point</p> <p>Complex</p> <p>Decimal</p> <p>Boolean Types</p> <p>Character Types</p> <p>Character String Types</p> <p>Design Issues</p> <p>Strings and Their Operations</p> <p>String Length Operations</p> <p>Evaluation</p> <p>Implementation of Character String Types</p> <p>User defined Ordinal types</p> <p>Enumeration types</p> <p>Designs</p> <p>Evaluation</p> <p>Subrange types</p> <p>Ada's design</p> <p>Evaluation</p> <p>Implementation of user defined ordinal types</p> <p>Array types</p> <p>Design issues</p> <p>Arrays and indices</p> <p>Subscript bindings and array categories</p> <p>Heterogeneous arrays</p> <p>Array initialization</p> <p>Array operations</p> <p>Rectangular and Jagged arrays</p> <p>Slices</p> <p>Evaluation</p>

	<ul style="list-style-type: none"> <li>Implementation of Array Types</li> <li>Associative Arrays <ul style="list-style-type: none"> <li>Structure and operations</li> <li>Implementing associative arrays</li> </ul> </li> <li>Record types <ul style="list-style-type: none"> <li>Definitions of records</li> <li>References to record fields</li> <li>Operations on records</li> <li>Evaluation</li> <li>Implementation of Record types</li> </ul> </li> <li>Union Types <ul style="list-style-type: none"> <li>Design issues</li> <li>Discriminated versus Free unions</li> <li>Evaluation</li> <li>Implementation of Union types</li> </ul> </li> <li>Pointer and Reference Types <ul style="list-style-type: none"> <li>Design issues</li> <li>Pointer operations</li> <li>Pointer problems <ul style="list-style-type: none"> <li>Dangling pointers</li> <li>Lost heap dynamic variables</li> </ul> </li> <li>Pointers in C and C++</li> <li>Reference types</li> <li>Evaluation</li> <li>Implementation of pointer and reference types <ul style="list-style-type: none"> <li>Representation of pointers and references</li> <li>Solution to dangling pointer problem</li> </ul> </li> <li>Heap management</li> </ul> </li> </ul>
<b>Unit - VI</b>	<p><b>Subroutines and Control Abstraction</b></p> <ul style="list-style-type: none"> <li>Fundamentals of Subprograms</li> <li>Design Issues for subprograms</li> <li>Local Referencing Environments</li> <li>Parameter-Passing Methods</li> <li>Parameters That are Subprograms</li> <li>Overloaded Subprograms</li> <li>Generic Subroutines <ul style="list-style-type: none"> <li>Generic Functions in C++</li> <li>Generic Methods in Java</li> </ul> </li> <li>Design Issues for Functions</li> <li>User-Defined Overloaded Operators</li> <li>Coroutines</li> <li>The General Semantics of Calls and Returns</li> <li>Implementing “Simple” Subprograms</li> <li>Implementing Subprograms with Stack-Dynamic Local Variables</li> <li>Nested Subprograms</li> <li>Blocks</li> <li>Implementing Dynamic Scoping</li> </ul>
<b>Unit - VII</b>	<p><b>Data Abstraction and Object Orientation</b></p> <ul style="list-style-type: none"> <li>Object-Oriented Programming</li> <li>Encapsulation and Inheritance</li> </ul>



	<p>Modules, Classes, Nesting (Inner Classes), Type Extensions, Extending without Inheritance</p> <p>Initialization and Finalization</p> <p>Choosing a Constructor, References and Values, Execution Order, Garbage Collection</p> <p>Dynamic Method Binding</p> <p>Virtual- and Non-Virtual Methods, Abstract Classes, Member Lookup, Polymorphism, Object Closures</p> <p>Multiple Inheritance</p> <p>Semantic Ambiguities, Replicated Inheritance, Shared Inheritance, Mix-In Inheritance</p>
<b>Unit - VIII</b>	<p><b>Concurrency</b></p> <p>Introduction</p> <p>Multiprocessor Architecture</p> <p>Categories of concurrency</p> <p>Motivations for studying concurrency</p> <p>Introduction to Subprogram-level concurrency</p> <p>Fundamental concepts</p> <p>Language Design for concurrency.</p> <p>Design Issues</p> <p>Semaphores</p> <p>Introduction</p> <p>Cooperation synchronization</p> <p>Competition Synchronization</p> <p>Evaluation</p> <p>Monitors</p> <p>Introduction</p> <p>Cooperation synchronization</p> <p>Competition Synchronization</p> <p>Evaluation</p> <p>Message Passing</p> <p>Introduction</p> <p>The concept of Synchronous Message Passing</p> <p>Java Threads</p> <p>The Thread class</p> <p>Priorities</p> <p>Competition Synchronization</p> <p>Cooperation Synchronization</p>

**Learning Resources:**

1. Scott, Programming Language Pragmatics, 3e(With CD) ISBN 9788131222560 Kaufmann Publishers, An Imprint of Elsevier, USA
2. Robert W. Sebesta, Concepts of Programming Languages, Eighth Edition, Pearson Education
3. Carl Townsend, Introduction to Turbo Prolog
4. Patrick Henry Winston & Berthold Klaus Paul Horn ,LISP 3rd edition –BPB
5. M. Gabbrielli, S. Martini, Programming Languages: Principles and Paradigms, Springer ISBN: 9781848829138

**CSC4207 - Networking and Information Security**  
**Credits: 4**

<b>Learning Outcomes</b>	<b>Suggested Pedagogical Processes</b>
To understand the architectural principles of computer networking.	Lecture method with use of ICT
To use routing algorithms while configuring a network.	Experiential learning through practical problem-solving
To write advanced Unix socket program for client server communication.	Practical demonstration and program assignment method
To get the knowledge of security – threats-detection and prevention method	Lecture method with use of ICT
To understand the various security measures related to computer and network security	Brainstorming

<b>Unit</b>	<b>Title and Contents</b>
I	<b>Review of Basic Concepts</b> Overview of TCP/IP Protocol Suite, wired & wireless LANS and WANS
II	<b>Routing Protocols</b> Forwarding Structure of a Router Routing Tables Intra – And Inter-Domain Routing RIP , OSPF , BGP
III	<b>The Internet Layer Protocols</b> Review of IPv4, IPv6 protocol, Transition from IPv4 to IPv6, ICMPv4, ICMPv6
IV	<b>The Transport Layer</b> The Transport Service Elements of Transport Protocols TCP, UDP frame formats
V	<b>Socket Introduction</b> Socket Address Structures (IPv4 & IPv6) Byte Ordering Functions, Byte Manipulation Functions readn, written, readline, isfdtype, getsockopt and setsockopt
VI	<b>Elementary TCP &amp; UDP Sockets</b> Network functions socket , connect, bind, listen, accept close, getsockname and getpeername TCP Server & client communication programs Concurrent Servers, Normal Startup & Termination of server and client, UDP Server & client communication programs

VII	<b>Introduction to security</b> The need for Security Overview of threats and attacks Security Approaches Principles of Security Types of Attacks
VIII	<b>Communication Security</b> Encryption-decryption model Plain text and cipher text ,Cryptography techniques Overview - Symmetric and Asymmetric key cryptography Steganography Types of network layer attacks Firewall (ACL, Packet Filtering, DMZ, Alerts and Audit Trails) IP security, Virtual Private Network, Intrusion

### Learning Resources

1. Behrouz A. Forouzan, TCP / IP Protocol Suite Fourth Edition
2. Andrew Tanenbaum, Computer Networks Fourth Edition
3. W. Richard Stevens, Bill Fenner, Andrew M. Rudoff, Unix Network Programming, Volume 1: The Sockets Networking API, 3/E PHI
4. Atul Kahate, Cryptography and Network Security Second Edition
5. Andrew Tanenbaum, Computer Networks Fourth Edition

**CSC4208 - Software Quality Assurance and Testing**  
**Credits: 4**

<b>Learning Outcomes</b>	<b>Suggested Pedagogical Processes</b>
Learn testing objectives and strategy and implement test cases.	Lecture method with use of ICT, algorithm assignment
Learn methods and procedures for software development that can scale up for large systems and that can be used to consistently produce high-quality software at low cost and with a small cycle time.	Lecture method and assignment
Learn systematic approach to the development, operation, maintenance of software.	Lecture method and assignment
Learn how to use available resources to develop software, reduce cost of software and how to maintain quality of software.	Lecture method with use Case Study
Learn methods of testing and maintenance of software's.	Lecture method with use of ICT

<b>Unit No.</b>	<b>Title of Unit and Contents</b>
I	<b>An introduction to Software Life Cycle from QA Engineer's Perspective</b> Development of a software and role of a QA . Various tools and techniques used by the IT industry for testing
II	<b>Software quality</b> Definition, Software errors, software faults and software failures, Software quality assurance – definition and objectives, Software quality assurance vs. software quality control, The objectives of SQA activities
III	<b>Pre-project SQA Components</b> Contract Review, Development and Quality Plan
IV	<b>SQA components in Project life cycle activities assessment</b> Verification and Validation., Various types of Reviews Inspections, Walkthrough, Software testing, Impact of CASE Tools
V	<b>SQA Infrastructure Components</b> Procedures and procedure manuals, Templates and Checklists Staff training, Corrective and preventive actions,

	Documentation control
VI	<b>Software Quality Factors</b> Mccall's Quality Model, Product, Process quality metrics
VII	<b>Standardization</b> ISO 9001 and ISO 9000-3 SEI-CMM IEEE 1012 standard ISO/IEC 12207 standard
VIII	<b>Configuration Management</b> Change control Release and version control Software configuration management audit
IX	<b>Quality Improvement Technique</b> Pareto Diagrams Cause-Effect Diagrams Scatter Diagrams Run Charts
X	<b>Quality Costs</b> Quality Cost Measurement Utilizing Quality Costs for Decision-Making
XI	<b>Case Studies</b>

### Learning Resources

1. Danial Galin, Software Quality Assurance from theory to implementation
2. Edwin Bennatan, Software Project management-
3. Roger S. Pressman, Software Engineering, TMH,7Th Ed.
4. Nina Godbole, Software Quality Assurance : Principles and Practices
5. Project Management Body of Knowledge – PMI
6. Donna C. S., Summers -Quality, 5th ed., Prentice-Hall, 2010.
7. Dale H., Besterfield -Total Quality Management, Prentice Hall, 2003.
8. John Wiley. J.F.Peters, Software engineering: An Engineering approach
9. [www.softwarecertifications.org](http://www.softwarecertifications.org)

**CSC4210 - Parallel Computing**  
**Credits: 4**

Learning Outcomes	Suggested Pedagogical Processes
Student will understand the Design paradigms of Parallel Computing and its model building process.	Lecture method with use of ICT
The student will get knowledge of parallel computing paradigms.	Lecture method with use of ICT
To develop the basic model with shared memory and check the performance.	Case studies and assignments method
To understand the practical experience of applying multi core programming.	Experiential learning through practical problem Solving and assignment

	Title and Contents
<b>Unit -I</b>	<b>Introduction to Parallel Computing</b> Why Parallel Computing & Scope of Parallel Computing, Sieve of Eratosthenes, Control and Data Approach, PRAM model of parallel computation, Design paradigms of Parallel Computing, examples, Bulk Synchronous Parallel (BSP) model.
<b>Unit -II</b>	<b>Classification</b> Flynn's Taxonomy, MPP, SMP, CC-NUMA, Clustering of Computers, Beowulf Cluster, Use of MPI in Cluster Computing. Debugging, Evaluating and tuning of Cluster Programs, Partitioning and Divide and Conquer Strategies. Cluster: dedicated high performance (HP), high availability (HA), CoPs, PoPs, CoWs; distributed, on-demand, high-throughput, collaborative, data-intensive computing, Interconnection networks.
<b>Unit –III</b>	<b>An overview of Parallel Programming Paradigms</b> Foster's design paradigm for Multi computing programming, Programmability Issues, Programming Models: Message passing, Message passing standards: PVM (Parallel Virtual Machine), MPI (Message Passing Interface) and its routines, Advanced Features of MPI
<b>Unit - IV</b>	<b>Overview of Programming with Shared Memory</b> Overview of Programming with Shared Memory: OpenMP (History, Overview, Programming Model, OpenMP Constructs, Performance Issues and examples, Explicit Parallelism: Advanced Features of OpenMP)
<b>Unit - V</b>	<b>Multi-Core programming</b> Multi-Core programming: Introduction to Multi cores Programming Software Multithreading using Tread Building Blocks (TBB) and Cilk++ programming, GPGPU programming with CUDA

## Learning Resources:

1. Quinn, M. J., Parallel Computing: Theory and Practice (McGraw-Hill Inc.).
2. Bary Wilkinson and Michael Allen: Parallel Programming Techniques using Networked of workstations and Parallel Computers, Prentice Hall, 1999.
3. R. Buyya (ed.) High Performance Cluster Computing: Programming and Applications, Prentice Hall, 1999.
4. William Gropp, Rusty Lusk, Tuning MPI Applications for Peak Performance, Pittsburgh (1996).
5. W. Gropp, E. Lusk, N. Doss, A. Skjellum, A high performance portable implementation of the message passing Interface (MPI) standard, Parallel Computing 22 (6), Sep 1996.
6. Gibbons, A., W. Rytter, Efficient Parallel Algorithms (Cambridge Uni. Press).
7. Shameem A and Jason, Multicore Programming, Intel Press, 2006.
8. CUDA Programming A Developer's Guide to Parallel Computing with GPUs Shane Cook, Morgan Kaufmann

**CSC4211 - Computer Science Practical - II**  
**Credits: 4**

<b>Learning Outcomes</b>	<b>Suggested Pedagogical Processes</b>
After completing assignments students will learn Hands-on C# .NET experience for professional advancement	

<b>Sr. No.</b>	<b>Title of Practical</b>
<b>Programming in C#</b>	
1	Working with String and String Builder and Arrays
2	Delegates & Exceptional Handling
3	Object Oriented Programming: Inheritance, Abstract classes and interfaces
4	Xml Handling and multithreading using Forms
5	Reflection
6	LINQ
<b>Full Stack –I - Web UI and Responsive UI Framework</b>	
1	Designing HTML5 Forms with new tags, attributes and audio visual support
2	Designing CSS3 for fonts, background images, text effect, animations and embedding media
3	Javascrpts - function, program, statements, objects
4	Bootstrap container , grid basics and tables
5	jQuery fundamentals and jQuery CSS.
6	JQuery Events and Effects



**CSC4212 - Project - II**  
**Credits: 4**

**Objective:**

**The objective of project is to make the students understand Requirement analysis, design and implementation cycle. Any open problem statement can be taken for implementation. The system can be designed in any programming language implemented in any platform.**

The Project can be platform, Language and technology independent. Project will be evaluated by project guide. Assessment will be done weekly in the respective batch. Evaluation will be on the basis of weekly progress of project work, progress report, oral, results and documentation and demonstration.

You should fill your status of the project work on the progress report and get the Signature of project guide regularly. Progress report should sharply focus how much time you have spent on specific task. (The format of progress report is given as follow.) You should keep all signed progress report. Project will not be accepted if progress report is not submitted and all responsibility remains with student.

**Project Progress Report**

<b>Roll No and Name of the Student</b>	
<b>Title of the project</b>	
<b>Project guide name</b>	

<b>Sr. No</b>	<b>From Date</b>	<b>To Date</b>	<b>Details of Project work</b>	<b>Project guide sign (with date)</b>

**Project Guide:**