Fergusson College (Autonomous)
Pune

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

With effect from June 2019

From Academic Year

2020-21

Program Outcomes (POs) for M. Sc. Programme	
PO1	**Disciplinary Knowledge:** Demonstrate comprehensive knowledge of the discipline that form a part of an postgraduate programme. Execute strong theoretical and practical understanding generated from the specific programme in the area of work.
PO2	**Critical Thinking and Problem solving:** Exhibit the skill of critical thinking and understand scientific texts and place scientific statements and themes in contexts and also evaluate them in terms of generic conventions. Identify the problem by observing the situation closely, take actions and apply lateral thinking and analytical skills to design the solutions.
PO3	**Social competence:** Exhibit thoughts and ideas effectively in writing and orally; communicate with others using appropriate media, build effective interactive and presenting skills to meet global competencies. Elicit views of others, present complex information in a clear and concise and help reach conclusion in group settings.
PO4	**Research-related skills and Scientific temper:** Infer scientific literature, build sense of enquiry and able to formulate, test, analyse, interpret and establish hypothesis and research questions; and to identify and consult relevant sources to find answers. Plan and write a research paper/project while emphasizing on academics and research ethics, scientific conduct and creating awareness about intellectual property rights and issues of plagiarism.
PO5	**Trans-disciplinary knowledge**: Create new conceptual, theoretical and methodological understanding that integrates and transcends beyond discipline-specific approaches to address a common problem.
PO6	**Personal and professional competence**: Performing independently and also collaboratively as a part of team to meet defined objectives and carry out work across interdisciplinary fields. Execute interpersonal relationships, self-motivation and adaptability skills and commit to professional ethics.
PO7	**Effective Citizenship and Ethics**: Demonstrate empathetic social concern and equity centred national development, and ability to act with an informed awareness of moral and ethical issues and commit to professional ethics and responsibility.
PO8	**Environment and Sustainability**: Understand the impact of the scientific solutions in societal and environmental contexts and demonstrate the knowledge of and need for sustainable development.
PO9	**Self-directed and Life-long learning:** Acquire the ability to engage in independent and life-long learning in the broadest context of socio-technological changes.

Program Specific Outcomes (PSOs) for M. Sc. Computer Science	
PSO No.	**Program Specific Outcomes (PSOs)** **Upon completion of this programme the student will be able to**
PSO1	**Academic competence**: **(i)** Understand a range of programming languages such as Python, R, C#, computing architecture, construction and design underlying in the field of computer science, and related disciplinary areas. **(ii)** Transfer the knowledge gained in the different computer science fields such as Soft computing, Information Security, Business intelligence to solve real world problems.
PSO2	**Personal and Professional Competence**: **(i)** Perform laboratory-orientated computer programs to demonstrate different programming language concepts. **(ii)** Design small executable software, useful mobile application by implementing analytical and critical thinking skills and deliver it on time in the form of report and PowerPoint presentation.
PSO3	**Research Competence:** **(i)** Analyse the computer software requirements and design a blueprint as well as executable application. **(ii)** Integrate the data collected during the software development process which enhances teamwork and leadership skills.
PSO4	**Entrepreneurial and Social competence:** **(i)** Apply skills learnt in the field of Artificial Intelligence, Data mining, Machine learning, Cloud Computing, Networking and Security, Software Quality Assurance in specific areas related to health, education, banking, defence etc. **(ii)** Develop skills required for social interaction.

Programme Structure

Year	Name of Paper	Course Code	Course Title	Credits
First Year (Semester - I)	Theory Core Paper - 1	CSC4101	Data Mining and Data Warehousing	4
	Theory Core Paper – 1	CSC4102	Analysis of Algorithms and Computing	4
	Theory Core Paper – 1	CSC4103	Operating System Internals	4
	Theory Core Paper – 1	CSC4104	Python Programming	4
	Practical Core Paper – 1	CSC4105	Computer Science Practical - I	4
	Practical Core Paper - 1	CSC4106	Project - I	4
First Year (Semester - II)	Theory Core Paper-1	CSC4201	Programming in C#	4
	Theory Core Paper-2	CSC4202	Full Stack-I: Web UI and Responsive UI Framework	4
	Theory Elective Paper- I	CSC4203	Business Intelligence and Analytics **OR**	4
		CSC4204	Robotics and Artificial Intelligence **OR**	4
		CSC4205	MOOC-I **OR**	4
		CSC4206	Principles of Programming Language (General Elective - I)	4
	Theory Elective Paper- II	CSC4207	Networking and Information Security **OR**	4
		CSC4208	Software Quality Assurance and Testing **OR**	4
		CSC4209	MOOC-II **OR**	4
		CSC4210	Parallel Computing (General Elective - II)	4
	Practical Core Paper- 1	CSC4211	Computer Science Practical - II	4
	Practical Core Paper - 2	CSC4212	Project - II	4

Second Year (Semester - III)	Theory Core Paper-1	CSC5301	Soft Computing	4
	Theory Core Paper-2	CSC5302	Full Stack-II : MEAN stack	4
	Theory Core Paper-3	CSC5303	Software Project Management	4
	Theory Elective Paper- III	CSC5304	Cloud Computing	4
		CSC5305	Machine Learning	4
		CSC5306	MOOC-III	4
		CSC5307	Modelling and Simulation (General Elective - III)	4
	Theory Elective Paper- IV	CSC5308	Cryptography and Cyber Forensics	4
		CSC5309	Digital Image Processing	4
		CSC5310	Data science using R	4
		CSC5311	MOOC-IV	4
		CSC5312	Natural Language Processing (General Elective - IV)	4
	Practical Core Paper-I	CSC5313	Project - III	4
Second Year (Semester - IV)		CSC5401	Industrial Training	8
	Total Credits			**80**

F.Y. M.Sc. Semester I		
Title of the Course and Course Code	**Data mining and Data Warehousing (CSC4101)**	**Number of Credits : 04**
Course Outcome (COs) **On completion of the course, the students will be able to:**		
CO1	Describe the mining architecture and the concept of algorithms.	
CO2	Explain the components of data warehousing architecture and mining algorithms.	
CO3	Implement various mining algorithms for the given data set by comparing the outcomes.	
CO4	Differentiate the outcomes of various algorithms and analyze the prediction process on the given dataset.	
CO5	Compare and contrast various algorithms with respect to efficiency and accuracy measures.	
CO6	Combine the data warehouse architecture and data mining algorithms and execute the predictive and analytical modelling system.	

Unit No.	Title of Unit and Contents
I	**Introduction to Data Mining** 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	**Introduction to Data Warehousing** 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	**Data Pre-processing** 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	**Data Mining Techniques** 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	**Classification & Prediction** 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	**Clustering** k-means, Expectation Maximization (EM) algorithm, Hierarchical clustering, Correlation clustering
VII	**Accuracy Measures** Precision, recall, F-measure, confusion matrix, cross-validation, bootstrap **Data Exploration** Types of Attributes Statistical Description of Data Data Visualization Measuring similarity and dissimilarity
VIII	**Brief overview of advanced techniques** 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

Title of the Course and Course Code	**Analysis of Algorithm and Computing (CSC4102)**	**Number of Credits : 04**
Course Outcome (COs) **On completion of the course, the students will be able to:**		
CO1	Describe the algorithmic strategies for different problems and the way to analyze them.	
CO2	Explain the importance and analysis of strategies using different examples.	
CO3	Implement various algorithms and analyze it using different possible strategies.	
CO4	Apply various algorithms to different type of problems.	
CO5	Compare and contrast various algorithms with respect to their complexities.	
CO6	Combine the knowledge of different strategy to write efficient algorithms for building an efficient software.	

Unit No.	Title of Unit and Contents
I	**Introduction** 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	**Design Strategies** Decrease and Conquer**:** Types, Insertion Sort, Topological Sorting Divide and Conquer: Concept, Sorting algorithms, Strassen's matrix multiplication
III	**Advanced Design and Analysis Techniques** 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	**Problem classification** 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 1
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

Title of the Course and Course Code	**Operating System Internals (CSC4103)**	**Number of Credits : 04**
Course Outcome (COs) **On completion of the course, the students will be able to:**		
CO1	Describe the Unix kernel architecture and concepts of files, users, groups, permissions.	
CO2	Discuss concepts related to buffers, inode, data structures related to file and process subsystems.	
CO3	Implement different file and process system calls using C programming language.	
CO4	Differentiate between Unix and Linux memory management system calls.	
CO5	Compare and contrast various signals available in Unix operating system.	
CO6	Combine file and process subsystems, memory and signal system calls to simulate the design of Unix operating system.	

Unit No.	Title of Unit and Contents
I	**General Overview** 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	**Introduction to the kernel** 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	**The buffer cache** 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	**Internal representation of file** 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	**File and Directory I/O** 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 fdatasync, fcntl, /dev/fd, stat, fstat, lstat, file types, fdatasync, 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
VI	**Process Environment, Process Control and Process Relationships** 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.
VII	**Memory Management** 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

VIII	**Signal Management** Signal concepts,signal function, unreliable function interrupted system calls, re-entrant functions, SIGCLD semantics, reliable-signal technology, kill and raise, alarm and pause, signal sets, sigprocmask, sigpending, sigsetjmp and siglongjmp, sigsuspend, abort, system function revisited, sleep, Basic Signal Management, Sending a Signal, Reentrancy, Signal Sets, Blocking Signals, Advanced Signal Management, Sending a Signal with a Payload

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

Title of the Course and Course Code	**Python Programming (CSC4104)**	**Number of Credits : 04**
Course Outcome (COs) **On completion of the course, the students will be able to:**		
CO1	Describe the various components of Python programming language.	
CO2	Grasp the idea of the individual constructs used in Python programming language.	
CO3	Implement the various concepts learnt in Python programming language.	
CO4	Differentiate between the other programming languages and Python.	
CO5	Compare and contrast the Python with the other programming languages.	
CO6	Combine all concepts in Python and develop a working application.	

Unit No.	Title of Unit and Contents
I	**Introduction to Python** Introduction Various IDES
II	**Data Types:** Numeric data types: int, float, complex string list and list slicing

	Tuple
III	**Control Flow, Functions, Modules and Packages** 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	**Data Structures** Lists as Stacks, Queues, Tuples and sequences Sets Dictionaries
V	**Python File Operation** 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	**Object Oriented Programming** 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	**Regular Expression** 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	**Database Interaction SQL** Database connection using python Creating and searching tables Reading and storing config information on database Programming using database connections
IX	**Image Processing with Google Vision API** Google's Vision API offerings of Vision API Vision API Client Library for Python
X	**Writing Web Crawler using Python**

Learning Resources:

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

Title of the Course and Course Code	**Computer Science Practical – I (CSC4105)**	**Number of Credits : 04**

Course Outcome (COs) On completion of the course, the students will be able to:	
CO1	Describe the various programming constructs of Python language.
CO2	Discuss the domain of the problem and analyze the problem to find the various entities of the domain.
CO3	Implement various programming constructs to code the analysis done for the problem domain.
CO4	Differentiate the analysis and the implementation phases appropriately.
CO5	Compare and contrast the use of various data structures to gain suitable knowledge about their implementation.
CO6	Combine all the programming constructs and develop a python program to give the desired functionality.

Sr. No.	Title of Experiment / Practical
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

Title of the Course and Course Code	**Project I (CSC4106)**	**Number of Credits : 04**
Course Outcome (COs) On completion of the course, the students will be able to:		
CO1	Describe a thorough and systematic understanding of project contents.	
CO2	Illustrate knowledge of computing and mathematics appropriate to the discipline.	
CO3	Implement project management knowledge, processes, tools and techniques in order to achieve success of the project.	
CO4	Design, implement, and evaluate a computational system to meet desired needs within realistic constraints.	
CO5	Integrate the identified modules using techniques and tools.	
CO6	Evaluate the generated modules using evaluation techniques and tools.	

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

Roll No and Name of the Student	
Title of the project	
Project guide name	

Sr. No	**From Date**	**To Date**	**Details of Project work**	**Project guide sign (with date)**

Project Guide

F.Y. M.Sc. Semester II		
Title of the Course and Course Code	**Programming in C# (CSC4201)**	**Number of Credits : 04**
Course Outcome (COs) **On completion of the course, the students will be able to:**		
CO1	State the fundamentals of Dot Net architecture and C# Programming language.	
CO2	Represent Object-Oriented principles applied in real life problems.	
CO3	Demonstrate the usage of Reflection and Dynamic programming features.	
CO4	Analyse the problem statement and choose appropriate features / technologies to provide solutions and communicate observations.	
CO5	Identify different approaches for building service-oriented applications.	
CO6	Evaluate appropriate data storage and data access strategy based on type, size of the data and need for data security.	

Unit No.	Title of Unit and Contents
I	**DOTNET Framework** 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	**Core C#** 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 Defining classes, class members, Interfaces, Properties, Access modifiers, properties and auto-implemented properties. Inheritance – Virtual Methods, polymorphism, hiding Methods, Abstract classes and methods, sealed classes and methods, is and as Operators Collections, Comparisons and Conversions Defining and using collections, Indexers, iterators, Type comparison, Value Comparison, Overloading Conversion operators Generics Using generics, Defining Generics, generic Interfaces, Generic methods, Generic Delegate
III	**Delegates, lambdas and events** Simple delegate, Multicast delegate, Action <T> and Func<T> delegate, Anonymous methods, Lambda Expressions Events: Event Publisher, Event Listener
IV	**Window Programming** Window Controls: Common Controls, Container Controls (Group box and Tab controls), Menus and Toolbars, Dialogs Deploying Window Application Click Once deployment, Introduction to WIX setup

V	**Data Access** File System Data XML Databases and Entity Framework Data Binding
VI	**LINQ** Overview, Standard Query Operators, LINQ with SQL, LINQ with Xml, LINQ with Collection
VII	**Reflection, Metadata and Dynamic Programming** Standard Attributes and Custom Attributes, System.Type class, Metadata, Reflection, Assembly class, Memberinfo class, Dynamic Type, dynamic vs var, Dynamic Language Runtime (DLR)
VIII	**Multithreading** Thread life cycle, Thread synchronization, Understanding Thread Pool, Background worker, introduction to Task, Async programming
IX	**Introduction to Windows Communication Foundation** WCF Architecture, WCF Application Components, WCF simple Programming, Data contracts, message contracts end points, type of binding one way, two ways, Hosting wcf in windows service, IIS. Introduction to REST using WCF

Learning Resources:

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

Title of the Course and Course Code	**Full Stack I – Web UI and Responsive UI Framework (CSC4202)**	**Number of Credits : 04**
Course Outcome (CO) On completion of the course, the students will be able to:		
CO1	Describe various HTML tags for designing simple Web pages.	
CO2	Illustrate and design interactive User Interface (UI) based web pages using functions of JavaScript.	
CO3	Implement different methods of CSS styling and bootstrap for creation of effective web pages.	
CO4	Analyze the responsiveness of a web page using jQuery.	
CO5	Implement JavaScript and jQuery methods to test and validate the web UI.	
CO6	Build real time, fast and dynamic web applications and web pages using AJAX.	

Unit No.	Title of Unit and Contents
I	**Overview of HTML5** History, Vision & 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.
II	**Introducing CSS3** 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.
III	**Overview of JavaScript** 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 & Iteration, Enumerating properties, Callbacks, JSON
IV	**Bootstrap** Introduction, Grid, Components, Plugins.

V	**jQuery and AJAX** 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.

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. https:// getbootstrap.com

Title of the Course and Course Code	**Business Intelligence and Analytics (CSC4203)**	**Number of Credits : 04**
Course Outcome (CO) On completion of the course, the students will be able to:		
CO1	Describe the Business architecture and the concepts of policies and procedures.	
CO2	Explain the composition of analytics and business policies with the specified strategy.	
CO3	Implement various procedural metrics in the given data set by comparing the outcomes.	
CO4	Differentiate between various business models and analytic applications to analyze business strategies.	
CO5	Compare and contrast various algorithms with respect to efficiency and accuracy measures related to business policies.	
CO6	Combine the strategies and cost evaluation techniques for analysing and predicting the business process model.	

Unit No.	Title of Unit and Contents
I	**Introduction to Business Intelligence** Definition and History of BI, Transaction processing versus analytical processing, BI implementation, Major tools and techniques of BI
II	**Data warehousing** 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	**Business performance management** Key performance indicators and operational metrics, Balanced scorecard , Six Sigma, Dashboards and scorecards
IV	**Introduction to Business Analytics** 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	**Basic Statistics in Business Analytics** 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	**Basics of Modeling** 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	**Data Visualization Techniques** Data Preparation and Exploration, Importance of data quality, Dealing with missing or incomplete data, Data Classification

VIII	**Modeling Techniques** Introduction to Common Modeling Techniques Cluster Analysis (Unsupervised Learning) Classification & Prediction (Supervised Learning) Classification - Training & Testing Sampling Data in Classification
IX	**Predictive analytics** Predictive modeling and Analysis - Regression Analysis, Multi-co-linearity, 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

Title of the Course and Course Code	**Robotics and Artificial Intelligence (CSC4204)**	**Number of Credits : 04**
Course Outcome (COs) **On completion of the course, the students will be able to:**		
CO1	Describe the knowledge of AI principles and techniques.	
CO2	Discuss the basic principles of AI toward problem solving.	
CO3	Implement various AI techniques by applying it to various problems.	
CO4	Differentiate between various AI techniques with the help of real problems.	
CO5	Compare and contrast various algorithms and their complexities.	
CO6	Combine the knowledge of AI techniques, Knowledge representation strategies, reasoning mechanisms to build an intelligent system.	

Unit No.	Title of Unit and Contents
I	**Introduction to Artificial Intelligence** What is AI? Early work in AI AI and related fields AI problems and Techniques
II	**Problems, Problem Spaces and Search** 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	**Heuristic Search Techniques** Generate-and-test Hill Climbing Best First Search Problem Reduction Constraint Satisfaction Mean-Ends Analysis

IV	**Knowledge Representation** 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	**Slot – and – Filler Structures** Weak Structures Semantic Networks Frames Strong Structures Conceptual Dependencies
VI	**Game Playing** Minimax Search Procedures Adding alpha-beta cutoffs Uncertianty Reasoning: Basic Probabilty Axioms, Baye's Rule, Baysian Classification, Certainty Factor Theory, Dempster Shafar Theory.
VII	**Learning** What is learning? Rote Learning Learning by taking advice Learning in problem solving Learning from examples Explanation based learning
VIII	**Robotics** 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.

Title of the Course and Course Code	**Principles of Programming Languages (CSC4206)**	**Number of Credits : 04**
Course Outcome (COs) **On completion of the course, the students will be able to:**		
CO1	Define Programming Language Spectrum.	
CO2	Compare various control flows and concurrency mechanisms.	
CO3	Classify / observe various features of a programming language with syntactical description.	
CO4	Analyse type disciplines in various programming languages.	
CO5	Evaluate the various strategies of data abstraction and control abstraction and implement in problem solving.	
CO6	Write small applications in non-imperative programming languages.	

Unit. No.	**Title of Unit and Contents**
	Introduction The Art of Language Design The Programming Language Spectrum Why Study Programming Languages? Compilation and Interpretation Programming Environments
	Non-Imperative Programming Models: Functional, Logic Languages 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

III	**Names, Scopes, and Bindings** 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, Object Closures Macro Expansion Turbo Prolog Introduction, facts, Objects and Predicates, Variables, Using Rules, Controlling execution fail and cut predicates
IV	**Control Flow** Expression Evaluation Precedence and Associativity, Assignments, Initialization, Ordering Within Expressions, Short-Circuit Evaluation Structured and Unstructured Flow Structured Alternatives to goto Sequencing Selection Short-Circuited Conditions, Case/Switch Statements Iteration Enumeration-Controlled Loops, Combination Loops, Iterators, Logically Controlled Loops Recursion Iteration and Recursion, Applicative and Normal -Order Evaluation

V	**Data Types** Introduction Primitive Data Types Numeric Types Integer Floating point Complex Decimal Boolean Types Character Types Character String Types Design Issues Strings and Their Operations String Length Operations Evaluation Implementation of Character String Types User defined Ordinal types Enumeration types Designs Evaluation Subrange types Ada's design Evaluation Implementation of user defined ordinal types Array types Design issues Arrays and indices Subscript bindings and array categories Heterogeneous arrays Array initialization Array operations Rectangular and Jagged arrays Slices Evaluation Implementation of Array Types Associative Arrays Structure and operations Implementing associative arrays Record types Definitions of records References to record fields Operations on records

	Evaluation Implementation of Record types Union Types Design issues Discriminated versus Free unions Evaluation Implementation of Union types Pointer and Reference Types Design issues Pointer operations Pointer problems Dangling pointers Lost heap dynamic variables Pointers in C and C++ Reference types Evaluation Implementation of pointer and reference types Representation of pointers and references Solution to dangling pointer problem Heap management
VI	**Subroutines and Control Abstraction** Fundamentals of Subprograms Design Issues for subprograms Local Referencing Environments Parameter-Passing Methods Parameters That are Subprograms Overloaded Subprograms Generic Subroutines Generic Functions in C++ Generic Methods in Java Design Issues for Functions User-Defined Overloaded Operators Coroutines The General Semantics of Calls and Returns Implementing “Simple” Subprograms Implementing Subprograms with Stack-Dynamic Local Variables Nested Subprograms Blocks Implementing Dynamic Scoping
VII	**Data Abstraction and Object Orientation** Object-Oriented Programming Encapsulation and Inheritance Modules, Classes, Nesting (Inner Classes), Type

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

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

Title of the Course and Course Code	**Networking and information Security (CSC4207)**	**Number of Credits : 04**
Course Outcome (COs) **On completion of the course, the students will be able to:**		
CO1	Describe the lower layer architecture of computer networks.	
CO2	Explain the concepts related to IP datagram and transport layer services.	
CO3	Apply the routing protocols to find the best suited route for forwarding the datagram in the network.	
CO4	Analyse the possible network attacks and propose the appropriate technique for securing the network system.	
CO5	Compare and contrast different cryptography types and techniques.	
CO6	Compile the knowledge of Unix socket system calls to build a client server communication application.	

Unit. No.	Title of Unit and Contents
I	**Review of Basic Concepts** Overview of TCP/IP Protocol Suite, wired & wireless LANS and WANS
II	**Routing Protocols** Forwarding Structure of a Router Routing Tables Intra – And Inter-Domain Routing RIP, OSPF, BGP
III	**The Internet Layer Protocols** Review of IPv4, IPv6 protocol, Transition from IPv4 to IPv6, ICMPv4, ICMPv6

IV	**The Transport Layer** The Transport Service Elements of Transport Protocols TCP, UDP frame formats
V	**Socket Introduction** Socket Address Structures (IPv4 & IPv6) Byte Ordering Functions, Byte Manipulation Functions readn, written, readline, isfdtype, getsockopt and setsockopt
VI	**Elementary TCP & UDP Sockets** 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	**Introduction to security** The need for Security Overview of threats and attacks Security Approaches Principles of Security Types of Attacks
VIII	**Communication Security** 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 Trials) 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

Title of the Course and Course Code	**Software Quality Assurance and Testing (CSC4208)**	**Number of Credits : 04**

Course Outcome (COs) On completion of the course, the students will be able to:	
CO1	Describe the software development life cycle from the perspective of Quality Assurance Engineer.
CO2	Illustrate various components of the project life cycle.
CO3	Implement various quality improvement techniques, assessment components and apply the suitable quality factors on the project.
CO4	Differentiate between the Quality Assurance and Testing.
CO5	Compare and contrast the different assessment components and various quality improvement techniques.
CO6	Combine all the techniques of quality control and improvement to assure quality of the project.

Unit No.	Title of Unit and Contents
I	**An introduction to Software Life Cycle from QA Engineer's Perspective** Development of a software and role of a QA. Various tools and techniques used by the IT industry for testing
II	**Software quality** 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	**Pre-project SQA Components** Contract Review, Development and Quality Plan
IV	**SQA components in Project life cycle activities assessment** Verification and Validation., Various types of Reviews Inspections, Walkthrough, Software testing, Impact of CASE Tools
V	**SQA Infrastructure Components** Procedures and procedure manuals, Templates and Checklists Staff training, Corrective and preventive actions, Documentation control
VI	**Software Quality Factors** Mccall's Quality Model, Product, Process quality metrics

VII	**Standardization** ISO 9001 and ISO 9000-3 SEI-CMM IEEE 1012 standard ISO/IEC 12207 standard
VIII	**Configuration Management** Change control Release and version control Software configuration management audit
IX	**Quality Improvement Technique** Pareto Diagrams Cause-Effect Diagrams Scatter Diagrams Run Charts
X	**Quality Costs** Quality Cost Measurement Utilizing Quality Costs for Decision-Making
XI	**Case Studies**

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

Title of the Course and Course Code	**Parallel Computing (CSC4210)**	**Number of Credits : 04**

Course Outcome (COs) **On completion of the course, the students will be able to:**	
CO1	Describe the importance and scope of parallel computing.
CO2	Understand various classification techniques and parallel programming paradigms.
CO3	Implement concepts of shared memory.
CO4	Differentiate the parallel programming with other traditional programming techniques.
CO5	Compare and Contrast parallel programming with shared memory to parallel programming with multi core.
CO6	Combine all the techniques together to develop the basic model with shared memory and check the performance and to understand the application of multicore programming.

Unit. No.	Title of Unit and Contents
I	**Introduction to Parallel Computing** 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.
II	**Classification** 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.
III	**An overview of Parallel Programming Paradigms** 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

IV	**Overview of Programming with Shared Memory** Overview of Programming with Shared Memory: OpenMP (History, Overview, Programming Model, OpenMP Constructs, Performance Issues and examples, Explicit Parallelism: Advanced Features of OpenMP)
V	**Multi-Core programming** 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

Title of the Course and Course Code	**Computer Science Practical – II (CSC4211)**	**Number of Credits : 04**

Course Outcome (COs) **On completion of the course, the students will be able to:**	
CO1	State fundamentals of Dot Net architecture, C# Programming language and User Interface (UI) designing.
CO2	Illustrate various Object-Oriented principles applied in real life problems. Demonstrate the use of CSS to design rich UI.
CO3	Design, develop professional console based and form based .NET applications. Design and implement various JavaScript methods to query a web page.
CO4	Differentiate various data storage and data access strategies. Apply various jQuery functions and methods to a Web application.
CO5	Test and validate the usage of Reflection and Dynamic programming features. Evaluate different events and effects associated with jQuery.
CO6	Combine features and programming constructs to build service-oriented applications and to develop web applications.

Sr. No.	Title of Experiment/ Practical
1	Working with String and String Builder
2	Arrays
3	Delegates: Single Cast and multicast
4	Exception Handling
5	Generic collections
6	Form based Application
7	Designing HTML5 Forms
8	Designing CSS3
9	ava Scripts
10	Multithreading
11	jQuery fundamentals and jQuery CSS
12	JQuery Events and Effects

Title of the Course and Course Code	**Project II (CSC4212)**	**Number of Credits : 04**
Course Outcome (CO) On completion of the course, the students will be able to:		
CO1	Describe a through and systematic understanding of project contents.	
CO2	Illustrate knowledge of computing and mathematics appropriate to the discipline.	
CO3	Implement project management knowledge, processes, tools and techniques in order to achieve project success.	
CO4	Design, implement, and evaluate a computational system to meet desired needs within realistic constraints.	
CO5	Integrate the identified modules using techniques and tools.	
CO6	Evaluate the generated modules using evaluation techniques and tools.	

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

Roll No and Name of the Student	
Title of the project	
Project guide name	

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

Project Guide: