Department Of Computer Science

Dr. Bhimrao Ambedkar University,

Agra

Detailed Syllabus

for

B.Sc. (Research)

and

M.Sc. (Computer Science)

Year wise Structure of
B.Sc. (Research) in Science
and
Master in Science (Computer Science)

Year	Sem.	Course Code	Paper Title	Theory/Practical	Credits
4	VII	B070701T	Compiler Design & Principles	Theory	4
		B070702T	Foundation of Artificial Intelligence	Theory	4
		B070703T	Software Engineering	Theory	4
			Lab	Practical	4
		B070901T	Information and Network Security	Theory	4
			Literature Survey	Research	4
4	VIII	B070801T	Digital Image Processing	Theory	4
		B070802T	Mobile Applications	Theory	4
		B070803T	Quantum Information and Computation	Theory	4
			Lab	Practical	4
		B071001T	Advance Design and Analysis of algorithms	Theory	4
		B070806R	Research Project	Research	4
		B070807T	One Minor Elective from Other Faculty	Theory	4
5	IX	Select any four theory papers			
		B070902T	Artificial Neural Networks	Theory	4
		B070903T	Machine Learning Techniques	Theory	4
		B070905T	Parallel Computing and Algoritms	Theory	4
		B070908T	Software Project Management	Theory	4
		-	Lab	Practical	4
		B070910T	Software Testing and Audit	Theory	4
		B070914T	Foundation on Data Science	Theory	4
		-	Literature Survey	Research	4
5	X	Select any four theory papers			
		B071002T	Quantum Neural network	Theory	4
		B071003T	Deep Learning & Pattern Recognition	Theory	4
		B071004T	Computer Vision	Theory	4
		B071009T	Principal of Software Reliability Engineering	Theory	4
		B071010T	IoT	Theory	4
		B071015T	Big Data & Data Analytics	Theory	4
		-	Lab	Practical	4
		-	Dissertation	Research	4
				Total Credits	**100**

Programme/Class: Bachelor(Research) in Science	Year: Fourth	Semester: VII
Subject: Computers		
Course Code: **B070701T**	Course Title: **Compiler Design & Principles**	

Course Outcomes

After the completion of the course, the student will be able to

CO 1-Acquire the basic knowledge of compiler, lexical rules, and grammars for programming language

CO 2- Apply parsing techniques on given expression, based on given grammar

CO 3-Describe and implement different techniques for intermediate code and machine code optimization to improve the program efficiency

CO 4- Describe and implement the use of symbol table, error detection and handling concept during different phases of compiler.

Credits: **4**	**Core Compulsory**
Max. Marks: **25+75**	Min. Passing Marks:

Total No. of Lectures-Tutorials-Practical (in hours per week): L-T-P: **4-0-0**

Unit	Topics	No. of Lectures
I	**Introduction and Lexical Analysis** Compiler, Translator and its need, the phases of a compiler, Cousins of the Compiler, grouping of Phases, Bootstrapping. Role of lexical analyzer, Input buffering, specification & Recognition of tokens, Regular sets and expression, Finite automata, Conversion of Regular expression to Finite automata, Obtaining Regular expression from Finite Automata, Optimization of Deterministic Finite automata states.	8
II	**Lexical Analysis** Lexical-analyzer generator, LEX-compiler, Formal grammars and their application to syntax analysis, BNF notation, ambiguity, YACC. The syntactic specification of programming languages: Context free grammars, derivation and parse trees.	7
III	**Basics of Parsing** Context Free Grammar, Derivation and Parse Tree, Basic Parsing Techniques: Parsers, Shift reduce parsing, operator precedence parsing, top down parsing, predictive parsers, Back tracking Parser or Recursive-descent parsing, LL parsing, Bottom Up Parsing (Shift –reduce parser, LR, Parser, SLR Parser, LALR Parser).	7

IV	**Construction of Parser** Automatic Construction of efficient Parsers: LR parsers, the canonical Collection of LR(0) items, constructing SLR parsing tables, constructing Canonical LR parsing tables, Constructing LALR parsing tables, using ambiguous grammars, an automatic parser generator, implementation of LR parsing tables.	8
V	**Syntax-directed Translation** Syntax-directed Translation schemes, & Implementation of Syntax-directed Translators, Intermediate code, postfix notation, Parse trees and syntax trees, L-attribute and S-attribute, three address code, quadruple, triples, Postfix notation, translation of assignment statements, Boolean expressions, statements that alter the flow of control, postfix translation, translation with a top down parser, Array references in arithmetic expressions, procedures call, declarations and case statements, implementation of syntax directed translator.	8
VI	**Symbol Tables** Contents of symbol table, Data structure for symbols tables, representing scope information, Run-Time Administration, Implementation of simple stack allocation scheme, storage allocation in block structured language, Storage allocation , Activation Record.	7
VII	**Error Detection & Recovery** Types of errors, Errors at different phases, Lexical Phase errors, syntactic phase errors, semantic errors, Error recovery strategies, Panic mode, Phrase level recovery, Error production, Global production, Error recovery in parsing, Run-time errors.	7
VIII	**Code Optimization and Code Generation** Principles sources of optimization, loop optimization, DAG representation of basic blocks, values numbers and algebraic laws, Global data-flow analysis, Machine-Independent Optimizations, Issues in the design of code generator, a simple code generator, Basic Blocks and Flow Graphs, Optimization of Basic Blocks, register allocation and assignment, code generation from DAG, Code Generator.	8

Suggested Readings:

1. Aho, Sethi & Ullman, "Compilers: Principles, Techniques and Tools", 2nd Edition, Pearson Education, 2007.
2. V Raghvan, " Principles of Compiler Design", Tata McGraw Hill Education, 2010.
3. Kenneth C. Louden," Compiler Construction", PWS Publishing Company (Cengage Learning),

Programme: **Master In Science (Computer Science)**	Year: Fourth	Semester: VII
Subject: Computer Science		
Course Code: **B070902T**	Course Title: **Foundation on Artificial Intelligence**	

Course outcomes: Upon the completion of the course, the student will be able to understand the basics of AI, its Applications in the real world, how to represent a real world problem (like Water Jug Problem, Travelling Salesman Problem, Tic Tac Toe, Chess Playing etc.) and to get the solution through various search algorithms. The student will learn, how machines answer to certain questions in various fields. Student can also understand about Expert systems that are used widely in various fields.

Credits: 2	Core Compulsory
Max. Marks: 25+75	Min. Passing Marks:

Total No. of Lectures-Tutorials-Practical (in hours per week): L-T-P: 2-0-0

Unit	Topics	No. of Lectures
I	Introduction To Artificial Intelligence, Foundations And History Of Artificial Intelligence, Problem Domain Of AI, General Issues In AI , AI Techniques, AI Tasks, Game Playing, Theorem Proving, Robotics, Perception And Speech Recognition, NLP, Expert System, Criteria Of Success, Level Of Modeling, State Space Representation, Problem Description. Applications Of Artificial Intelligence, Intelligent agents	7
II	Problem Representation, Introduction To Search : Searching For Solutions, Production system, control strategies, Problems like water jug, 8-puzzle, travelling salesman and etc., Back tracking algorithm,Breadth First Search, Depth First Search, Iterative Deepings, Problem Characteristic, Commutative Production System, Random search, Bidirectional search, Uniform cost searching, branch and bound searching.	8
III	Heuristic Search Methods, A* Algorithm, Observation on A* algorithm, admissibility of A*, Problem Reduction, And-OR Graphs, Hill Climbing, Constraint Satisfaction, Game Playing, Minmax Search Procedure And Alpha Beta Cutoff, Local beam search, Memory based searching, Simulated annealing.	7
IV	Knowledge Representation Issues In Knowledge Representation Characteristic Of The Knowledge And	8

	Knowledge Representation Model, Representation Mapping, Issues, Various Kind Of Knowledge Representation Models, First Order Predicate Logic, Its Properties, Representation In Wff Application Of Predicate Logic In A.I, Backward Reasoning Method, Resolution, Rules Of Inference, Modus Ponens, Clause Form Representation, Theorem Proving, Control Strategies (BF, Linear Input Form, Set Of Support Etc.), Unification, Questioning And Answering.	
V	Natural deduction, Rule Based Systems, Deficiencies In Clause Form And Resolution, Forward Rule Base Deduction System, Backward Rule Base Deduction System, Representation Of Facts, Rule And Goal Wffs In AND OR Graph Representation, Unify Composition And Answer Extraction. Expert Systems, Components Of Expert Systems, Applications Of Expert System	7
VI	Object - Centred Structure Of Knowledge Representation, Its Advantages, Isa And Instance Representation, Class Inclusion And Membership, Property Inheritance, Semantic Net, Partition Semantic Net, Presentation Of Wffs Of Predicate Logic In Semantic Net, Frame Structure, Regular Class And Meta Classes, Property Inheritance Algorithm. Scripts, conceptual dependency.	8
VII	Handling Uncertainty , Basic probability theory, prior probability, conditional probability, inference using full joint distribution, Bay's rule, Probabilistic Reasoning, Bayesian Networks, Exact Inference In Bayesian networks, Inference by enumeration, Using Of Certainty Factory, Different Models For Handling Uncertainty And Its Reasoning For A.I., Case Study Of MYCIN	8
VIII	Learning, forms of learning, inductive learning, learning decision trees, ensemble learning, logical formulation of learning, knowledge in learning, explanation based learning, learning using relevance information, inductive logic programming.	7

Suggested Readings:

1. Elaine Rich and Kevin Knight, "Artificial Intelligence", McGraw-Hill, India, 2017, Third Edition,
2. Dan W. Patterson, "Artificial Intelligence and Expert Systems", Prentice Hall of India,2015

3. Stuart Russell, Peter Norvig, "Artificial Intelligence – A Modern Approach", Pearson Education, 2010, Third Edition

4. N.P.Padhy, "Artificial Intelligence and Intelligent Systems",Oxford University Press, USA, 2005

This course can be opted as an elective by the students of following subjects:

B.Sc. In Electronics, B.Sc. In Physics, B.Sc. In Statistics, B.Sc. In Mathematics, B.Sc. In Engineering, B.Sc. Vocational, BCA, Bachelor in Fine Arts. B.E. , B.Tech, B.A. (Maths), B.A.(Phil) with knowledge of mathematics up to class 12'th, B.A.(Psychology) knowledge of mathematics up to class 12'th.

Suggested Continuous Evaluation Methods:

2 Periodical Tests(each of 7.5 marks) + one seminar (5 marks) + 3 marks of assignment + 2 marks of Class Interaction

Course prerequisites:

To study this course, a student must have had the subject Mathematics in class12th and elementary knowledge of any Computer Programming Language.

Suggested equivalent online courses:

http://ugcmoocs.inflibnet.ac.in/ugcmoocs/view_module_pg.php/1484

Further Suggestions:

List of Open Source Software/learning website:

1. http://www.journals.elsevier.com/artificial-intelligence/
2. https://www.technologyreview.com/s/534871/our-fear-of-artificial-intelligence/
3. http://www.senfoundry.com/artificial-intelligence-mcqs-inductive-logic-unification-lifting-1/

Programme/Class: **B.Sc. (Research) In Science**	Year: **Fourth**	Semester: **Seven**
Subject: **Computer Science**		
Course Code: **B0707031**	Course Title: **Software Engineering**	

Course outcomes:

CO1 Describe software engineering layered technology and process framework.
CO2 Introduces theories, models, and techniques that provide a basis for the software development life cycle.
CO3 Introduces software testing approaches including verification and validation, static analysis, reviews, inspections, and audits.
CO4 Understanding of the role of project management including planning, scheduling, risk management, etc.
CO5 Work as an individual and/or in team to develop and deliver quality software.

Credits: **4**	**Core Compulsory**
Max. Marks: **25+75**	Min. Passing Marks:

Total No. of Lectures-Tutorials-Practical (in hours per week): **4-0-0**

Unit	Topic	No. of Lectures
I	**Software Engineering Fundamentals:** Definition of Software, Software characteristics, Software Applications. Software Process: Software Process Models - Waterfall model, prototyping model, spiral model, Incremental model, concurrent development model. Project management Concepts: The Management Spectrum - The People, The Product, The Process, The Project.	11

II	**Software Process and Project Metrics :** Measures , Metrics and Indicators , Software measurement Size - Oriented Metrics , Function - Oriented Metrics , Extended Function point metrics	4
III	**Software Project Planning:** Project Planning Objectives, Software Project Estimation , Decomposition Techniques - Problem Based Estimation Process Based Estimation ,Empirical Estimation Models- The COCOMO Model Risk Analysis and Management: Software risks, Risk identification, Risk Projection, Risk Refinement, Risk Mitigation , Monitoring and Management.	11
IV	**Software Quality Assurance:** Basic concepts- Quality, Quality Control, Quality Assurance, Cost of Quality , Software Quality Assurance (SQA) , Formal Technical Review	4
V	**Software Configuration Management** Baselines , Software Configuration Items, The SCM Process, Version Control, Change Control, Configuration Audit, Status Reporting. Analysis Concepts and Principles: Requirements Elicitation for Software, Analysis Principles. The Information Domain, Modeling, Partitioning, Essential and Implementation Views, Specification: Specification Principles, Representation, The Software Requirement Specification (SRS)	8
VI	**Design Concepts and Principles:** Design Principles, Design Concepts – Abstraction, Refinement, Modularity, Software Architecture, Control Hierarchy, Structural Partitioning, Data Structure. Software Procedure, Structure, Information Hiding, Effective Modular Design- Cohesion, Coupling	7
VII	**Software Testing :**Testing Objectives & principles, Unit Testing, Integration Testing (Top Down Integration , Bottom Up Integration, Regression Testing, Smoke Testing), Validation Testing (Alpha and Beta Testing), System Testing (Recovery Testing, Security Testing, Stress Testing, Performance Testing).	7
VIII	**UNIT-V Reengineering:** Software Reengineering, Reverse Engineering, Restructuring, Forward Engineering CASE Tools: What is CASE, Building Blocks of CASE, A Taxonomy of CASE Tools, Integrated CASE Environments, The Integration Architecture, The CASE Repository.	8

Suggested Readings:

1. Roger S.Pressman, Software engineering- A practitioner's Approach, McGraw-Hill
2. Ian Sommerville, Software engineering, Pearson education Asia, 6th edition, 2000.
3. Pankaj Jalote- An Integrated Approach to Software Engineering, Springer Verlag, 1997.
4. James F Peters and Witold Pedrycz, "Software Engineering – An Engineering Approach",

Programme/Class: **M.Sc. (Computer Science)**	Year: Fourth	Semester: VII
Subject: **Computer Science**		
Course Code: **B070901T**	Course Title: **Information and Network Security**	

Course outcomes: After the completion of the course, the students will be able:

CO1: To understand the concepts of information security and their need and application.

CO2: To understand the network security services and mechanisms.

CO3: To apply cryptographic algorithms for information and network security.

CO4: To learn the concept of key, key management, key distribution in cryptographic systems.

CO5: To understand Data integrity, Authentication, Digital Signatures Biometric Security Systems.

Credits: 4	**Core Compulsory**
Max. Marks: **25+75**	Min. Passing Marks:

Total No. of Lectures-Tutorials-Practical (in hours per week): L-T-P: **4-0-0**

Unit	Topics	No. of Lectures
I	Information security, Information Management Technologies, Security policies, Policy enforcement & related issues, Components of Information System, Security Models, Balancing Information Security and Access, Cipher Model, Stream ciphers and block ciphers, Cryptography, Cryptanalysis, Attacks, Substitution and Transposition techniques,Web Security threats, Internet Security Protocols.	8
II	Symmetric and asymmetric key cryptography, Symmetric key Ciphers: DES structure, DES Analysis, Security of DES, variants of DES, Multiple encryption and triple DES, Electronic Code Book, Block cipher modes of operation, Cipher Block Chaining Mode, AES structure, Analysis of AES.	8
III	Asymmetric key Ciphers, Random number generation, Fundamentals of entity authentication, Zero-knowledge mechanisms, Cryptographic Protocols, Authentication and key establishment protocols, Principles of public key cryptosystems, Public Key Cryptosystems with Applications, Requirements and Cryptanalysis, RSA algorithm, its computational aspects and security.	8
IV	Cryptographic MAC and Hash Functions, their applications, Simple hash functions, its requirements and security, Hash functions based on Cipher Block Chaining, Secure Hash Algorithm (SHA), Whirlpool, HMAC.	7
V	Key management fundamentals, Key lengths and lifetimes, Key generation, Key establishment, Key storage, Key	8

	Usage, Governing key management, Public-Key Management, Certification of public keys, certificate lifecycle, Public-key management models, Key distribution, symmetric key distribution, Diffie-Hillman Key Exchange algorithm, Man-In-Middle attack.	
VI	Digital Signature, its properties, requirements and security, various digital signature schemes (Elgamal and Schnorr), NIST digital signature algorithm, Defining Intrusion Detection, Security concepts intrusion Detection concept, determining strategies for Intrusion Detection, Responses, Vulnerability Analysis, Credentialed approaches, Technical issues.	7
VII	Remote user authentication with symmetric and asymmetric encryption, Kerberos, IPSec, Secure Socket Layer (SSL), Transport Layer Security (TLS), Secure Electronic Transaction (SET), Pretty Good Privacy (PGP), S/MIME.	7
VIII	Biometric Fundamentals, Types of Biometrics, Fingerprints and Hand Geometry, Facial and Voice Recognition, Iris and Retina scanning, Signature Recognition and Keystroke Dynamics, Behavioral and Esoteric Biometric Technologies, Issues involving Biometrics, Privacy, Policy and Legal Concerns Raised by Biometrics.	7

Suggested Readings:

1. William Stallings, "Cryptography And Network Security: Principles and Practice," Sixth Edition, Pearson Education, 2013.
2. Mark Stamp, "Information Security Principles and Practice," Willy India, 2006.
3. Forouzan and Mukhopadhyay, "Cryptography & Network Security," Second Edition, McGrawHill Education, 2010.
4. Atul Kahate, "Cryptography and Network Security," Fourth Edition, McGrawHill, 2019.
5. C K Shyamala, N Harini, T R Padmanabhan, "Cryptography and Security," Wiley-India, 2011.
6. Godbole, "Information Systems Security: Security Management, Metrics, Frameworks and Best Practices," Second Edition, Wiley, 2017.

This course can be opted as an elective by the students of following subjects:

B.Sc. In Electronics, B.Sc. In Physics, B.Sc. In Statistics, B. Sc. Mathematics, B.Sc. In Engineering, B.Sc. Vocational, BCA, B.E./B.Tech, M.E. / M. Tech

Suggested Continuous Evaluation Methods:

2 Periodical Tests(each of 5 marks) + 10 marks for the submission of any two programs written in any programming language from the given list + 9 marks of assignment + 2 marks of attendance.

Course prerequisites:

Mathematical concepts including number theory, random numbers, and basic concepts of computer networks and communication

Programme/Class: **B.Sc. (Research) In Science**	Year: **Fourth**	Semester: VIII
Subject: **Computer Science**		
Course Code: B070801T	Course Title: **Digital Image Processing**	

Course outcomes:
The student will be able to understand the basics of Computer Graphics, he/she will be able to do certain operations of graphics such as drawing different shapes, editing of these shapes. The student would be able to do 2D and 3D Transformations like translation, scaling, rotation, reflection and many more.
The student will be able to understand the basics of Digital Image processing, he/she will be able to perform transformations on images to enhance the quality of these images. The student would be able to understand about various filters that can be applied on Images to enhance an image or to restore that image. He/she would be able to detect a point, a line or an edge from the images and he/she would also understand the several techniques to compress an image.

Credits: **4**	**Core Compulsory**
Max. Marks: **25+75**	Min. Passing Marks:

Total No. of Lectures-Tutorials-Practical (in hours per week): L-T-P: **4-0-0**

Unit	Topics	No. of Lectures
I	**Introduction**: Pixel, Frame, Buffer, Applications Of Computer Graphics, Graphic Displays- Random Scan Displays, Raster Scan Displays, Points And Lines, Line Drawing Algorithms, Circle Generating Algorithms, Polygon Generation And Polygon Filling Algorithm	6
II	**2DTransformations**: Translation, Scaling, Rotation, Reflection, Homogeneous Coordinates , Matrix Representations, Composite Transformations, Reflections And Shearing. **Three Dimensional**: 3-D Geometric Primitives, 3-D Object	8

	Representation, 3-D Transformation: : Translation, Scaling, Rotation, 3-D Viewing, Projections, 3-D Clipping.	
III	**Windowing And Clipping**: 2-D Clipping Algorithms- Line Clipping Algorithms Like Cohen Sutherland Line Clipping Algorithm, Liang Barsky Algorithm, Polygon Clipping – Sutherland Hodgeman Polygon Clipping, Text Clipping	6
IV	**DIGITAL IMAGE FUNDAMENTALS**: Applications, Steps in Digital Image Processing – Components of Digital Image Processor, Image Acquisition **IMAGE DIGITIZATION**: Image Sampling and Quantization, Representing Digital Images, Spatial and Gray level resolution, Zooming and Shrinking, Relationships between pixels: neighbors of a pixel, Adjacency, Connectivity, Regions, Boundaries, Color Image fundamentals : RGB	8
V	**IMAGE ENHANCEMENT IN SPATIAL DOMAIN**: Gray level transformations, Histogram processing: Histogram Equalization, Histogram Matching, Basics of Spatial Filtering, Smoothing and Sharpening Spatial Filtering **IMAGE ENHANCEMENT IN FREQUENCY DOMAIN**: Introduction to Fourier Transform, DFT, DCT, Walsh Hadamard, Smoothing and Sharpening frequency domain filters : Ideal, Butterworth and Gaussian filters(low pass and high pass filters)	8
VI	**IMAGERESTORATION** : Image Restoration , degradation model, Properties, Noise models , Mean Filters, Order Statistics, Adaptive filters, Band reject Filters, Band pass Filters, Notch Filter, Inverse Filtering , Wiener Filtering	8
VII	**IMAGE SEGMENTATION**: Point detection, Line detection, Edge detection, Edge linking via Hough transform , Region based segmentation ,Morphological processing- erosion and dilation, Segmentation by morphological watersheds	8
VIII	**IMAGE COMPRESSION AND RECOGNITION**: Need for data compression, Lossy compression: Transform coding, Lossless Compression: Huffman, Run Length Encoding, Arithmetic coding, JPEG standard, MPEG, Fidelity criteria.	8

Programme/Class: **B.Sc. (Research) In Science**	Year: **Fourth**	Semester: **Eighth**
Subject: **Computer Science**		
Course Code: **B070802T**	Course Title: **Mobile Applications**	

Course outcomes:

After the completion of the course the students will be able to:

1. Understands the basic concepts of event driven programming.
2. Design and implement mobile applications.
3. Understand data persistence.
4. Perform Remote Data-Storage and Communication.

Credits: **4**	**Core Compulsory**
Max. Marks: **25+75**	Min. Passing Marks:

Total No. of Lectures-Tutorials-Practical (in hours per week): **4-0-0**

Unit	Topic	No. of Lectures
I	**Event Driven Programming:** UI event loop, Threading for background tasks, Outlets / actions, delegation, notification, Model View Controller (MVC) design pattern.	7
II	**Mobile application issues:** limited resources (memory, display, network, file system), input / output (multi-touch and gestures), sensors (camera, compass, accelerometer, GPS)	8
III	**Development tools:** Apple IOS toolchain: Objective-C, Xcode IDE, Interface Builder, Device simulator.	7

IV	**Frameworks**: Objective-C and Foundation Frameworks, Cocoa Touch, UIKit, Others: Core Graphics, Core Animation, Core Location and Maps, Basic Interaction.	8
V	**Common UI's for mobile devices**: Navigation Controllers, Tab Bars, Table Views, Modal views, UI Layout.	7
VI	**Data Persistence**: Maintaining state between application invocations, File system, Property Lists, SQLite, Core Data	8
VII	**Remote Data Storage and Communication:** "Back End" / server side of application, RESTful programming, HTTP get, post, put, delete, database design, server side JavaScript / JSON	8
VIII	**Code signing:** security, Keychain, Developers and App Store License Agreement	7

Suggested Readings:

1. Rajiv Ramnath, Roger Crawfis, and Paolo Sivilotti, Android SDK 3 for Dummies, Wiley, 2011.
2. Valentino Lee, Heather Schneider, and Robbie Schell, Mobile Applications: Architecture, Design, and Development, Prentice Hall, 2004.
3. Brian Fling, Mobile Design and Development, O'Reilly Media, 2009. Maximilliano
4. Firtman, Programming the Mobile Web, O'Reilly Media, 2010.
5. Christian Crumlish and Erin Malone, Designing Social Interfaces, O'Reilly Media, 2009.

This course can be opted as an elective by the students of following subjects:
B. Sc In Engineering, BCA, MCA, M.Sc.(IT)

Suggested Continuous Evaluation Methods: **Max. Marks: 25**

1. Assessment Type: Class Tests (Max. Marks 14)
2. Assessment Type: Quizzes/ Objective Tests / Recognition Type (such as MCQs; True or False; Matching; Classifying) /Recall Type -Filling Blanks; One word / Phrase Answers (Max Marks: 5)
3. Assessment Type: Assignments (Max Marks: 4)
4. Assessment Type: Class Interaction (Max. marks: 2)

Course prerequisites:
To study this course, a student must have had the subject **Data Structures, DBMS, Operating System, Object Oriented Programming with C++**

Suggested equivalent online courses:
1. https://onlinecourses.nptel.ac.in/noc20_cs52/preview
2. https://nptel.ac.in/courses/106/106/106106156/

Programme/Class: **B.Sc. (Research) in Science**	Year: **Fourth**	Semester: **Eighth**
Subject: **Computer Science**		
Course Code: 0070803T	Course Title: **Quantum Information & Computation**	

Course outcomes:

Students would learn the framework of quantum computation, and how that may be useful for future quantum technologies. This course teaches the fundamentals of quantum information processing, including quantum computation, quantum cryptography, and quantum information theory. The last 20 years have seen the discovery of algorithms that directly harness the laws of quantum mechanics to speed up certain computations and ensure secrecy of communications. There are fast quantum algorithms to factor large integers and compute discrete logarithms, which, if implemented, threaten the security of the encryption schemes in common use today. This possibility has spurred several major and ongoing attempts to build quantum computers. Quantum computation might also be useful in simulating complex quantum systems such as large molecules.

Course Objective: The main objective of this course is to provide the student with the basic understanding of quantum computation and quantum information. Following objectives will cover:

- Understanding of quantum bits and quantum gates
- Analyze the behavior of basic quantum algorithms
- Implement simple quantum algorithms and information channels in the quantum circuit model
- Simulate a simple quantum error-correcting code
- Prove basic facts about quantum information channels

This course will primarily focus on the mathematical and computer science aspect of it. It will start the by answering "why quantum computing?" and then move on to study the basic linear algebra and computer science needed to understand the theory of quantum computation. Then it will explore the idea of quantum circuit model in which most of the quantum algorithms are designed. The final part of the course will look at quantum algorithms and advantage they offer over classical computer.

Credits: 4	**Core Compulsory**
Max. Marks: 25+75	Min. Passing Marks:

Total No. of Lectures-Tutorials-Practical (in hours per week): L-T-P: 4-0-0

Unit	Topics	No. of Lectures
I	Introduction to Quantum computing: History of quantum computation and quantum information, quantum bits, general view of quantum computation, quantum circuits, algorithms, operations,Qubits versus classical bits, Bloch sphere representation of a qubit, multiple qubits	8
II	Background Mathematics and Physics: Hilber space, Bases and linear independence, Linear operators and matrices, Pauli matrices, inner products, Eigenvectors and Eigen values,Adjoint and Hermitian operators, Tensor product, operator functions,	7
III	Postulates of quantum mechanics: State space, Evolution, quantum measurement, Distinguishing quantum states, projective measurement, phase, composite system, density operator, EPR and the Bell inequality	7
IV	Quantum Circuits: single qubit gates, multiple qubit gates, design of quantum circuits, Quantum algorithms, single qubit operations, controlled operations, measurement, universal quantum gates, quantum circuit model for computation, simulation of quantum systems	8
V	Quantum Information and Cryptography: Comparison between classical and quantum information theory. Bell states. Quantum teleportation. Quantum Cryptography, no cloning theorem.	8
VI	Quantum Algorithms: Classical computation on quantum computers. Relationship between quantum and classical complexity classes. Deutsch's algorithm, Deutsch-Jozsa algorithm, Shor factorization, Grover search.	7
VII	Noise and error correction: classical noise and Markov process, quantum operations, Axiomatic approach to quantum operations, examples of quantum noise and quantum operations, application of quantum operations	8

VIII	Quantum error correction: Introduction, three qubit bit flip code, three qubit phase flip code, the Shor code, Discretization of the errors, Independent error models, Degenerate codes, the quantum Hamming bound, classical linear code, calderbank-Shore-Steane codes	7

Suggested Readings:

1. Quntum Computation and Quantum Information: Michael A. Nielsen and Isaac L. Chuang, Cambridge University Press, 2010
2. Benenti G., Casati G. and Strini G., Principles of Quantum Computation and Information, Vol. I: Basic Concepts, Vol II: Basic Tools and Special Topics, World Scientific. 2004
3. 3 Pittenger A. O., An Introduction to Quantum Computing Algorithms 2000

This course can be opted as an elective by the students of following subjects:

B.Sc. in Electronics, B.Sc. in Physics, B.Sc. in Statistics, B. Sc. Mathematics, B.Sc. in Engineering, B.Sc. Vocational, BCA, B.E./B.Tech, M.E. / M. Tech

Suggested Continuous Evaluation Methods:

2 Periodical Tests(each of 5 marks) + 10 marks for the submission of any two programs written in any programming language from the given list + 3 marks of assignment + 2 marks of attendance.

Course prerequisites:

There are no formal prerequisites for this course. Informally, student should be familiar with calculus and linear algebra, and know some probability and discrete math. Knowledge of quantum mechanics is NOT a prerequisite; quantum concepts will be introduced as needed. Similarly, knowledge of algorithms and complexity are not prerequisites either; these also will be introduced as needed.

Suggested equivalent online courses:

Learning website: https://learn.xpro.mit.edu/quantum-computing,
https://www.coursera.org/learn/quantum-computing-algorithms,
https://www.coursera.org/projects/programming-quantum-computer-qiskit

List of experiments using Qiskit library

1. Implement the multi-qubits and show the various quantum operations.
2. Implement the Quantum circuit for preparing the Bell state.
3. Implement the different quantum gates and show the outcomes.
4. Implement the measurement of state.
5. Implement the Deutsch's algorithm.
6. Implement the Deutsch's-Jozsa algorithm
7. Implement the algorithm of Shor factorization.
8. Implement the Grover search.
9. Create a system of a single qubit in the state |0>. and using it to create a QuantumCircuit using the Identity operator.
10. Implement the Perfect Coin Algorithm.

Programme: Bachelor (Research) in Science	Year: Fourth	Semester: VIII
Subject: Computer Science		
Course Code: B071001T	Course Title: Advance Design and Analysis of Algorithms	
Course Outcomes:		
Credits:	Core Compulsory	
Max Marks: 25+75	Min Passing Marks:	

Total No. of Lectures – Tutorial – Practical (in Hours per week): L-T-P: 4:0:0

Unit	Topics	No. of Lectures
I	Introduction: Algorithms, Analyzing Algorithms, Complexity of Algorithms, Growth of Functions, Performance Measurements, Sorting and Order Statistics - Shell Sort, Quick Sort, Merge Sort, Heap Sort, Comparison of Sorting Algorithms, Sorting in Linear Time	08
II	Advanced Data Structures: Red-Black Trees, B – Trees, Binomial Heaps, Fibonacci Heaps, Tries, Skip List	08
III	Divide and Conquer with Examples Such as Sorting, Matrix Multiplication, Convex Hull and Searching. Greedy Methods with Examples Such as Optimal Reliability Allocation, Knapsack, Minimum Spanning Trees – Prim's and Kruskal's Algorithms, Single Source Shortest Paths - Dijkastra's and Bellman Ford Algorithms.	08
IV	Dynamic Programming with Examples Such as Knapsack. All Pair Shortest Paths – Warshal's and Floyd's Algorithms, Resource Allocation Problem. Backtracking, Branch and Bound with Examples Such as Travelling Salesman Problem, Graph Coloring, n-Queen Problem, Hamiltonian Cycles and Sum of Subsets.	08
V	Selected Topics: Algebraic Computation, Fast Fourier Transform, String Matching, Theory of NP Completeness, Approximation Algorithms and Randomized Algorithms	08

Suggested Readings:

1. Thomas H. Coreman, Charles E. Leiserson and Ronald L. Rivest, "Introduction to Algorithms", Printice Hall of India.
2. E. Horowitz & S Sahni, "Fundamentals of Computer Algorithms",
3. Aho, Hopcraft, Ullman, "The Design and Analysis of Computer Algorithms" Pearson Education, 2008.
4. LEE "Design & Analysis of Algorithms (POD)",McGraw Hill
5. Richard E.Neapolitan "Foundations of Algorithms" Jones & Bartlett Learning
6. Jon Kleinberg and Éva Tardos, Algorithm Design, Pearson, 2005.
7. Michael T Goodrich and Roberto Tamassia, Algorithm Design: Foundations, Analysis, and Internet Examples, Second Edition, Wiley, 2006.
8. Harry R. Lewis and Larry Denenberg, Data Structures and Their Algorithms, Harper Collins, 1997 9. Robert Sedgewick and Kevin Wayne, Algorithms, fourth edition, Addison Wesley, 2011.
10. Harsh Bhasin,"Algorithm Design and Analysis",First Edition,Oxford University Press.
11. Gilles Brassard and Paul Bratley, Algorithmics:Theory and Practice, Prentice Hall,1995

This course can be opted as an elective by the students of following subjects:

B.Sc. in Electronics, B.Sc. in Physics, B.Sc. in Statistics, B.Sc. in Engineering, B.Sc. Vocational, BCA, Bachelor in Fine Arts., B.E./ B.Tech.

Suggested Continuous Evaluation Methods: **Max.Marks:25**
1. Assessment Type: Class Tests (Max. Marks14)
2. Assessment Type: Quizzes / Objective Tests/ Recognition Type (such as MCQs; True or False; Matching; Classifying) / Recall Type- Filling Blanks; One word / Phrase Answers (Max Marks:5)
3.Assessment Type: Assignments (Max Marks:4)
4.Assessment Type: Class Interaction (Max Marks:2)

Course prerequisites:
To study this course, a student must have had the subject Software Engineering

Suggested equivalent online courses

Further Suggestions:
None

Programme/Class: **Bachelor(Research) In Science**	Year: Fifth	Semester: IX
Subject: **Computer Science**		
Course Code: B070501T	Course Title: **Artificial Neural Networks**	

Course outcomes:

1. Get the exposure to Artificial Neural Networks.
2. Understand the Modeling of Neuron and Express both Artificial Intelligence and Neural Network
3. Analyze ANN learning, Error correction learning, Memory-based learning, Hebbian learning, Competitive learning and Boltzmann learning
4. Implement Simple Perceptron, Perceptron learning algorithm, Convergence theorem, liner classifier and limitation of perceptron architecture
5. Develop feed forward multilayer neural network, Develop Delta learning rule of the output layer and Radial basis network
6. Implementation of Recurrent neural networks, Analysis of Hopfield energy function and problem of local minima.
7. Implementation of stochastic Hopfield neural network, simulated annealing and Boltzmann machine.
8. Get the exposure of Self organizing Map, ART and Necognitron

Credits: 4	**Core Compulsory**
Max. Marks: 25+75	Min. Passing Marks:

Total No. of Lectures-Tutorials-Practical (in hours per week): L-T-P: 4-0-0

Unit	Topics	No. of Lectures
I	Introduction to Neural Networks: Neural Network, Human Brain, Pattern and data, pattern recognition tasks, Models of Neuron, Neural networks viewed as directed graphs, Biological Neural Network, Artificial neuron, Artificial Neural Network architecture, ANN learning, analysis and applications, Topology of artificial neural networks.	7

II	Activation and synaptic dynamics: Activation dynamics model, Bidirectional associative memory, Lyapunov function analysis for stability, fixed point stability, Grossberg activation models, Synaptic dynamics models, learning equation, types of learning, requirements of learning laws, Learning methods (Hebbian learning, Competitive learning, Error correction learning, Reinforcement learning)	8
III	Linear associator, Supervised Hebbian learning and its analysis, Single layer Perception, Pattern classification, Linear classifier, Simple Perceptron, Perceptron learning algorithm, Convergence Theorem and Limitation of Perception.	7
IV	Feed forward ANN, Structures of Multi-layer feed forward networks. Back propagation algorithm, Back propagation - training and convergence, Functional approximation with back propagation. Practical and design issues of back propagation learning	8
V	Radial Basis Function Networks, Pattern separability and interpolation, Regularization Theor Regularization and RBF networks.RBF network design and training. Approximation properties of RBF.	7
VI	Feedback neural networks: Pattern storage and association, Hopfield model, Energy analysis of Hopfield network, Problem of false minima, Stochastic networks, Equilibrium of stochastic networks, Stability in stochastic networks, operation of a stochastic network, simulated annealing, Architecture of a Boltzmann machine, Boltzmann learning law	8
VII	Competitive Learning neural networks: Introduction, Components of competitive learning networks, Basic competitive Learning, Pattern Clustering, linear Vector Quantization, Analysis of feature mapping network, Self organizing map	7
VIII	Classical ART Network, Simplified ART Architecture,ART1 and ART2 Architecture and algorithms, Applications, Sensitivities of ordering of data. Applications of ANN : Pattern classification – Recognition of Olympic games symbols, Recognition of printed Characters. Neocognitron – Recognition of handwritten characters. NET Talk: to convert English text to speech. Recognition of consonant vowel (CV) segments, texture classification and segmentation	8

Suggested Readings:

1. B. Yegnanarayana"ARTIFICIAL NEURAL NETWORK" PHI Publication, 1998.
2. "Fundamentals of artificial neural networks",MIT press, Mohamad H. Hassoun , 1995
3. Kevin L. Priddy, Paul E. Keller,"Artificial neural networks: An Introduction" - SPIE Press, 2005
4. Nelson, Morgan, "Artificial neural network: Electronic Implementations" – IEEE Press, 1990

This course can be opted as an elective by the students of following subjects:

B.Sc. In Electronics, B.Sc. In Physics, B.Sc. In Statistics, B. Sc. Mathematics, B.Sc. In Engineering, B.Sc. Vocational, BCA, B.E./B.Tech, M.E. / M. Tech

Suggested Continuous Evaluation Methods: **Max. Marks: 25**

1. Assessment Type: Class Tests (Max. Marks 14)
2. Assessment Type: Quizzes/ Objective Tests / Recognition Type (such as MCQs; True or False; Matching; Classifying) /Recall Type -Filling Blanks; One word / Phrase Answers (Max Marks: 5)
3. Assessment Type: Assignments (Max Marks: 4)
4. Assessment Type: Class Interaction (Max. marks: 2)

Course prerequisites:

Higher Engineering Mathematics e.g. linear algebra, multivariate calculus and Probability theory, Fundamental knowledge of signals and systems along with types, Mathematical representation of signals and system modeling in time as well as frequency domain. Transforms especially like Laplacian, Fourier and Z. Artificial Intelligence and Control system Engineering.

Suggested equivalent online courses:

Learning website: www.ocw.mit.edu, www.learnartificialneuralnetworks.com, www.neural-forecasting.com

Programme/Class: **Master in Science (Computer Science)**	Year: **Fifth**	Semester: IX
Subject: **Computer Science**		
Course Code: **B07090T**	Course Title: **Machine Learning Techniques**	

Course outcomes: The students will be able to select and implement machine learning techniques and computing environment that are suitable for the applications under consideration., he will be able to solve problems associated with batch learning and online learning. Students will have the ability to understand and apply scaling up machine learning techniques and associated computing techniques and technologies. He would be able to recognize and implement various ways of selecting suitable model parameters for different machine learning techniques.

Credits: **4**	Elective
Max. Marks: **25+75**	Min. Passing Marks:

Total No. of Lectures-Tutorials-Practical (in hours per week): L-T-P: **4-0-0**

Unit	Topics	No. of Lectures
I	**Introduction** Class overview: Class organization, topics overview, Introduction: What is ML; Problems, Definition of learning systems. Goals and applications of machine learning. Aspects of developing a learning system: training data, concept representation, function approximation.	6

II	**Linear regression**; SSE; gradient descent; closed form; normal equations; features, Overfitting and complexity; training, validation, test data, **Classification problems**; decision boundaries; nearest neighbor methods. Probability and classification, Bayes optimal decisions, Naive Bayes and Gaussian class-conditional distribution **Linear classifiers**, Bayes' Rule and Naive Bayes Model, **Logistic regression,** online gradient descent	7
III	**Decision Tree Learning** Representing concepts as decision trees. Recursive induction of decision trees. Picking the best splitting attribute: entropy and information gain. Searching for simple trees and computational complexity. Overfitting, noisy data, and pruning.	8
IV	**Ensemble Learning** Bagging, boosting, and DECORATE. Active learning with ensembles. **Experimental Evaluation of Learning Algorithms** Measuring the accuracy of learned hypotheses. Comparing learning algorithms: cross-validation, learning curves, and statistical hypothesis testing.	7
V	**Computational Learning Theory:** Models of learnability: learning in the limit; probably approximately correct (PAC) learning. Sample complexity: quantifying the number of examples needed to PAC learn. Computational complexity of training. Sample complexity for finite hypothesis spaces. PAC results for learning conjunctions, kDNF, and kCNF. Sample complexity for infinite hypothesis spaces, Vapnik-Chervonenkis dimension	8
VI	**Support Vector Machines** Kernels for learning non-linear functions. **Bayesian Learning** Probability theory and Bayes rule. Naive Bayes learning algorithm. Parameter smoothing. Generative vs. discriminative training. Logistic regression. Bayes nets and Markov nets for representing dependencies. k-Nearest-neighbor algorithm. Case-based learning. **Text Classification:** Bag of words representation. Vector space model Relevance feedback and Rocchio algorithm. Versions of nearest neighbor and Naive Bayes for text.	8

VII	**Clustering and Unsupervised Learning** Learning from unclassified data. Clustering. Hierarchical Aglomerative Clustering. k-means partitional clustering. Expectation maximization (EM) for soft clustering. Semi-supervised learning with EM using labeled and unlabled data.	8
VIII	**Language Learning** Classification problems in language: word-sense disambiguation, sequence labeling. Hidden Markov models (HMM's). Veterbi algorithm for determining most-probable state sequences. Forward-backward EM algorithm for training the parameters of HMM's. Use of HMM's for speech recognition, part-of-speech tagging, and information extraction.	8

Suggested Readings:

1. Tom M. Mitchell, "Machine Learning" , McGraw-Hill Education (India) Private Limited, 2013.

2. Ethem Alpaydin, "Introduction to Machine Learning " , The MIT Press 2004.

3. Stephen Marsland, "Machine Learning: An Algorithmic Perspective", CRC Press, 2009.

4. Dutt Saikat , "Machine Learning" ,Pearson

This course can be opted as an elective by the students of following subjects:

B.Sc. In Electronics, B.Sc. In Physics, B.Sc. In Statistics, B.Sc. In Engineering, B.Sc. Vocational, BCA, Bachelor in Fine Arts., B.E./B.Tech

Suggested Continuous Evaluation Methods:

2 Periodical Tests(each of 5 marks) + 10 marks for the submission of any two programs written in any programming language from the given list + 3 marks of assignment + 2 marks of attendance.

Course prerequisites:

To study this course, a student must have had the subject Mathematics in class12th.

Suggested equivalent online courses:

Further Suggestions:

Programs:

Programme/Class: **M.Sc. (Computer science)**	Year: **Fifth**	Semester: IX
B0709027 Subject: **Computer Science**		
Course Code:	Course Title: **Parallel Computing and Algorithms**	

Course outcomes:

After the completion of the course the students will be able to:

1. Understands the difference between sequential and parallel mode.
2. Understands the parallel programming platforms.
3. Write parallel algorithm for different computational models.
4. Understand parallel algorithms for different data structures.

Credits: **4**	**Core Compulsory**
Max. Marks: **25+75**	Min. Passing Marks:

Total No. of Lectures-Tutorials-Practical (in hours per week): **4-0-0**

Unit	Topic	No. of Lectures
I	**Introduction to Parallel Computing:** Sequential model, need of alternative model, Motivating Parallelism, Scope of Parallel Computing.	4
II	**Parallel Programming Platforms:** Implicit Parallelism: Trends in Microprocessor Architectures, Limitations of Memory System Performance, Dichotomy of Parallel Computing Platforms, Physical Organization of Parallel Platforms, Communication Costs in Parallel Machines	8
III	**Parallel computational models:** PRAM, LMCC, Hypercube, Cube Connected Cycle, Butterfly, Perfect Shuffle Computers, Tree model, Pyramid model, Fully Connected model, PRAM-CREW, EREW models, simulation of one model from another one.	10
IV	**Performance Metrics:** Performance Measures of Parallel Algorithms, speed-up and efficiency of PA, Cost optimality, An example of illustrate Cost-optimal algorithms- such as summation, Min/Max on various models.	8
	Parallel Sorting Networks: Parallel Merging Algorithms on CREW/EREW/MCC, parallel Sorting Networks on	7

V	CREW/EREW/MCC, linear array	
VI	**Parallel Searching Algorithm**: Kth element, Kth element in X+Y on PRAM, Parallel Matrix Transportation and Multiplication Algorithm on PRAM, MCC, Vector-Matrix Multiplication, Solution of Linear Equation, Root finding.	8
VII	**Graph Algorithms**:- Definitions and Representation, Minimum Spanning Tree: Prim's Algorithm, Single-Source Shortest Paths: Dijkstra's Algorithm, All-Pairs Shortest Paths, Transitive Closure, Connected components.	7
VIII	**Search Algorithms for Discrete Optimization Problems:** Definitions and Examples, Sequential Search Algorithms, Search Overhead Factor, Parallel Depth-First Search, Parallel Best-First Search, Speedup Anomalies in Parallel Search Algorithms	8

Suggested Readings:

1. M.J. Quinn, "Designing Efficient Algorithms for Parallel Computer" by Mc Graw Hill.
2. S.G. Akl, "Design and Analysis of Parallel Algorithms"
3. S.G. Akl, "Parallel Sorting Algorithm" by Academic Press.

This course can be opted as an elective by the students of following subjects:
B. Sc In Engineering, BCA, MCA, M.Sc.(IT)

Suggested Continuous Evaluation Methods: **Max. Marks: 25**

1. Assessment Type: Class Tests (Max. Marks 14)
2. Assessment Type: Quizzes/ Objective Tests / Recognition Type (such as MCQs; True or False; Matching; Classifying) /Recall Type -Filling Blanks; One word / Phrase Answers (Max Marks: 5)
3. Assessment Type: Assignments (Max Marks: 4)
4. Assessment Type: Class Interaction (Max. marks: 2)

Course prerequisites:
To study this course, a student must have had the subject Data Structures, Algorithm Design and Analysis, Computer Network, Computer Architecture,

Suggested equivalent online courses:

1. https://nptel.ac.in/courses/106/102/106102114/
2. https://www.coursera.org/learn/introduction-high-performence-computing

Further Suggestions:
None

Programme/Class: M.Sc (Computer Science)	Year: Fifth	Semester: IX

Subject: **Computer Science**

Course Code: **B0709081**	Course Title: **Software Project Management**

Course outcomes:

1. Apply the process to be followed in the SDLC models.
2. Able to understand communication, modeling, construction & deployment practices in software development.
3. Understand the concepts of various software testing methods.
4. Explain the quality management & different types of metrics used in software development.
5. Apply the concepts of project management & planning.

Credits: 4	**Specialization Group B: Core Compulsory**
Max. Marks: **25+75**	Min. Passing Marks:

Total No. of Lectures-Tutorials-Practical (in hours per week): **4-0-0**

Unit	Topic	No. of Lectures
I	**Introduction To Software Project Management:** Introduction, What Is a Project? Software Projects Versus Other Types of Project, Contact Management and Technical Project Management, Activities Covered by Software Project Management, Plans, Methods, and Methodologies, Some Ways of Categorizing Software Projects, What Is Management?, Problems with Software Projects, Setting Objectives, Stakeholders, The Business Case, Requirement Specification, Management Control, Overview of Project Planning (Step wise)	8
II	**Project Evaluation & Selection Of An Appropriate Project Approach:** Introduction, Strategic Assessment, Technical Assessment, Cost-Benefit Analysis, Cash Flow Forecasting, Cost-Benefit Evaluation Techniques, Risk Evaluation. Selection Of An Appropriate Project Approach: Introduction, Choosing Technologies, Technical Plan Contents List, Choice of Process Models, Structure Versus Speed of Delivery, The Waterfall Model, The V-Process Model, The Spiral Model, Software Prototyping, Other ways of Categorizing Prototyping, Controlling Changes during Prototyping, Incremental Delivery, Dynamic Systems Development Method, Extreme Programming, Managing Iterative Processes.	7
III	**Software Effort Estimation:** Introduction, Where are Estimates done? , Problems with Over-and Under- Estimates, The Basis for Software Estimating, Software Effort Estimation Techniques, Expert Judgement, Estimating by Analogy, Albrecht Function Point Analysis, Function Point Mark II, Object Points, A Procedural Code-Oriented Approach, COCOMO: A Parametric Model.	8
IV	**Activity Planning:** Introduction, The Objectives of Activity Planning, When to Plan, Project Schedules, Projects and Activities, Sequencing and Scheduling Activities, Network Planning Models, Formulating a Network Model, Adding the Time Dimension, The Forward Pass, The Backward Pass, Identifying the Critical Path, Activity Float, Shortening the Project Duration, Identifying Critical Activities, Activity-On-Arrow Networks.	7

V	**Risk Management & Resource Allocation**: Introduction, The Nature of Risk, Types of Risk, Managing Risk, Hazard Identification, Hazard Analysis, Risk Planning and Control, Evaluating Risks to the Schedule.Resource Allocation: Introduction, The Nature of Resources, Identifying Resources Requirements, Scheduling Resources, Creating Critical Paths, Counting the Cost, Being Specific, Publishing the Resources Schedule, Cost Schedules, The Scheduling Sequence.	7
VI	**Monitoring, Control & Managing Contracts**: Introduction, Creating the Framework, Collecting the Data, Visualizing Progress, Cost Monitoring, Earned Value, Prioritizing Monitoring, Getting the Project Back to Target, Change Control. Managing Contracts: Introduction, Types of Contract, Stages in Contract Placement, Typical Terms of a Contract, Contract Management, Acceptance.	8
VII	**Managing People And Organizing Teams**: Introduction, Understanding Behaviour, Organizational Behaviour: A Background, Selecting The Right Person For The Job, Instruction In The Best Methods, Motivation, The Oldham- Hackman Job Characteristics Model, Working In Groups, Becoming A Team, Decision Making, Leadership, Organizational Structures.	7
VIII	**Software Quality**: Introduction, The Place Of Software Quality In Project Planning, The Importance Of Software Quality, Defining Software Quality, ISO 9126, Practical Software Quality Measures, Product Versus Process Quality Management, External Standards, Techniques To Help Enhance Software Quality, Quality Plans.	8

Suggested Readings:

1. B.Nuges and M.Cotierell- Software Project Management 3rd Edn, TMH, New Delhi, 2004.

2. P.Jolote- Software Project Management in Practice, Pearson Education, New Delhi, 2002.

This course can be opted as an elective by the students of following subjects:
B. Sc In Engineering, BCA, MCA, M.Sc.(IT)

Suggested Continuous Evaluation Methods: **Max. Marks: 25**

1. Assessment Type: Class Tests (Max. Marks 14)
2. Assessment Type: Quizzes/ Objective Tests / Recognition Type (such as MCQs; True or False; Matching; Classifying) /Recall Type -Filling Blanks; One word / Phrase Answers (Max Marks: 5)
3. Assessment Type: Assignments (Max Marks: 4)
4. Assessment Type: Class Interaction (Max. marks: 2)

Course prerequisites:
To study this course, a student must have had the subject Software Engineering

Suggested equivalent online courses:

1. https://onlinecourses.nptel.ac.in/noc19_cs70/preview
2. https://nptel.ac.in/courses/106/105/106105218/
3. https://www.classcentral.com/course/swayam-software-project-management-14294

Further Suggestions:
None

Programme/Class: **M.Sc. (Computer science)**	Year: **Fifth**	Semester: **Ninth**
Subject: **Computer Science**		
Course Code: **B0709101**	Course Title: **Software Testing and Audit**	

Course outcomes:

1. To understand Software Engineering, Testing Process, Terminologies in Testing, SRS
2. To understand different types of software testing (i.e. Functional Testing, Structural Testing)
3. To apply different types of testing with tools
4. To understand different types of Software Testing Activities (i.e Levels of Testing)
5. To understand Object Oriented Testing
6. To understand Testing Web Applications

Credits: **4**	**Specialization Group B: Elective**
Max. Marks: **25+75**	Min. Passing Marks:

Total No. of Lectures-Tutorials-Practical (in hours per week): **4-0-0**

Unit	Topic	No. of Lectures
I	**Review of Software Engineering:** Overview of Software Evolution, SDLC, Testing Process, Terminologies in Testing: Error, Fault, Failure, Verification, Validation, Difference Between Verification and Validation, Test Cases, Testing Suite, Test Oracles, Impracticality of Testing All Data; Impracticality of Testing AllPaths.	8
II	**Verification:** Verification Methods, SRS Verification, Source Code Reviews, User Documentation Verification, Software, Project Audit, Tailoring Software Quality Assurance Program by Reviews, Walkthrough, Inspection and Configuration Audits.	7
III	**Functional Testing:** Boundary Value Analysis, Equivalence Class Testing, Decision Table Based Testing, Cause Effect Graphing Technique. **Structural Testing:** Control Flow Testing, Path Testing, Independent Paths, Generation of Graph from Program, Identification of Independent Paths, Cyclomatic Complexity, Data Flow Testing, Mutation Testing	8
IV	**Regression Testing:** What is Regression Testing? Regression Test cases selection, Reducing the number of test cases, Code coverage prioritization technique. **Reducing the number of test cases:** Prioritization guidelines, Priority category, Scheme, Risk Analysis.	7
V	**Software Testing Activities:** Levels of Testing, Debugging, Testing techniques and their applicability, Exploratory Testing	7

VI	**Automated Test Data Generation:** Test Data, Approaches to test data generation, test data generation using genetic algorithm, Test Data Generation Tools, Software Testing Tools, and Software test Plan.	8
VII	**Object Oriented Testing:** Definition, Issues, Class Testing, Object Oriented Integration and System Testing.	7
VIII	**Testing Web Applications:** Web Testing, User Interface Testing, Usability Testing, Security Testing, Performance Testing, Database testing, Post Deployment Testing.	8

Suggested Readings:

1 Yogesh Singh, "Software Testing", Cambridge University Press, New York, 2012

2. K..K. Aggarwal & Yogesh Singh, "Software Engineering", New Age International Publishers, New Delhi, 2003.

3. Roger S. Pressman, "Software Engineering – A Practitioner's Approach", Fifth Edition, McGraw-Hill International Edition, New Delhi,2001.

4. Marc Roper, "Software Testing", McGraw-Hill Book Co., London, 1994.

5. M.G. Trivedi, Software Testing & Audit, Khanna Publishing House

6. Boris Beizer, "Software System Testing and Quality Assurance", Van Nostrand Reinhold, New York, 1984.

This course can be opted as an elective by the students of following subjects:
B. Sc In Engineering, BCA, MCA, M.Sc.(IT)

Suggested Continuous Evaluation Methods: **Max. Marks: 25**

1. Assessment Type: Class Tests (Max. Marks 14)

2. Assessment Type: Quizzes/ Objective Tests / Recognition Type (such as MCQs; True or False; Matching; Classifying) /Recall Type -Filling Blanks; One word / Phrase Answers (Max Marks: 5)

3. Assessment Type: Assignments (Max Marks: 4)

4. Assessment Type: Class Interaction (Max. marks: 2)

Course prerequisites:
To study this course, a student must have had the subject Software Engineering

Suggested equivalent online courses:
1. https://onlinecourses.nptel.ac.in/noc19_cs71/preview
2. https://onlinecourses.nptel.ac.in/noc20_cs19/preview
3. https://www.classcentral.com/course/swayam-software-testing-14295

Further Suggestions:
None

Programme: M.Sc. (Computer Science)	Year: Fifth	Semester: IX

Subject: Computer Science

Course Code: B070914T	Course Title: Foundation of Data Science

Course Outcomes:

Credits:	Core Compulsory
Max Marks: 25+75	Min Passing Marks:

Total No. of Lectures – Tutorial – Practical (in Hours per week): L-T-P: 4:0:0

Unit	Topics	No. of Lectures
I	**Data visualization:** Elements of data visualization, Exploration plots: Scatterplots, Line plots, barplots, boxplots, Advanced plots: correlation, regression, biplots, Reporting using visualization Keywords: seaborn, plotly. **Data preparation:** Handling missing data: imputation methods , Feature transformation and engineering. • Keywords: sklearn	08
II	Supervised learning: **Linear models for regression:** Linear models and non-linear feature maps, Model evaluation, Bias-Variance tradeoff, Penalized regression, Cross validation and model selection **Linear models for classification:** Logistic regression , Misclassification, ROC, AUC, Class imbalance **Non-linear models: decision trees:** Decision trees, Variable selection, Random Forests, Bagging and boosting, Keywords: sklearn, linear models, cross validation, regularization, lasso, trees, ensembles, boosting	08
III	Unsupervised learning: Clustering, PCA and SVD, Keywords: clustering, PCA	08
IV	STATISTICS FOR DATA SCIENCE: Introduction to Statistics, Harnessing Data, Exploratory Analysis, Distributions, Hypothesis & Computational Techniques, Correlation & Regression	08

V	**PYTHON FOR DATA SCIENCE**: Introduction to Data Science with Python, Python Basics: Basic Syntax, Data Structures, Data Objects, Math, Comparison Operators, Condition Statements, loops, lists, tuples, dicts, functions, Numpy Package, Pandas Package, Exploratory Data Analysis: Data Cleaning, Data Wrangling, Exploratory Data Analysis: Case Study **VISUAL ANALYTICS FOUNDATION**: Visual Analytics Basics, Basic Charts, Plots	08

Suggested Readings:

1. James, G., Witten, D. , Hastie, T. and Tibshirani, R., An introduction to Statistical Leanings with applications in R. Springer, 2013.
2. Hastie, T. , Tibshirani, R. and Friedman, J., The Elements of Statistical Learning. Springer, 2009.
3. Bishop, C.M. Pattern Recognition and Machine Learning. Springer, 2006

This course can be opted as an elective by the students of following subjects:

B.Sc. in Electronics, B.Sc. in Physics, B.Sc. in Statistics, B.Sc. in Engineering, B.Sc. Vocational, BCA, Bachelor in Fine Arts., B.E./ B.Tech.

Suggested Continuous Evaluation Methods: **Max.Marks:25** .
1. Assessment Type: Class Tests (Max. Marks14)
2. Assessment Type: Quizzes / Objective Tests/ Recognition Type (such as MCQs; True or False; Matching; Classifying) / Recall Type- Filling Blanks; One word / Phrase Answers (Max Marks:5)
3.Assessment Type: Assignments (Max Marks:4)
4.Assessment Type: Class Interaction (Max Marks:2)

Course prerequisites:
To study this course, a student must have had the subject Software Engineering

Suggested equivalent online courses

Further Suggestions:
None

Programme/Class: **Master In Science (Computer Science)**	Year: **Fifth**	Semester: **X**
Subject: **Computer Science**		
Course Code: **B071002T**	Course Title: **Quantum Neural Networks**	

Course outcomes:
Students would learn the framework of quantum neural networks, and how that may be useful for future machine intelligence technologies.This course teaches the fundamentals of quantum neural networks, including quantum computation, quantum gates, and entanglement with quantum states.There are fast quantum algorithms to factor large integers, compute discrete logarithms, and iterative process for operator construction which, if implemented, threaten the pattern recognitiontask. This possibility has spurred several major and ongoing attempts to build quantum computers. Quantum computation might also be useful in simulating complex quantum systems such as large molecules.

Credits: **4**	**Elective**
Max. Marks: **25+75**	Min. Passing Marks:

Total No. of Lectures-Tutorials-Practical (in hours per week): L-T-P: **4-0-0**

Unit	Topics	No. of Lectures
I	What is Quantum, Quantum Computation, Quantum Algorithms, Quantum Information Processing, Principles of Quantum Computing, Postulates of Quantum Computing, Quantum Machine Learning (QML), Why QML?, Building Blocks of QML: Qubits, Superposition, Interference, Entanglement etc, Inherent Parallelism of Quantum Computing, Applications of QML.	8
II	Quantum Neural Networks (QNN), Why QNN? , Neural Computing, Quantum Computing, Neural Networks: Towards Quantum Analogs, How Pattern Recognition leads us to QNN, Many Universe Approach, Quantum Associative Memory, Classical Neural Networks vs Quantum Associative Memory, Implementation of QNN: Physicial realizations and challenges, Can QNN outperform Classical ANN ? Review of existing approaches to QNNs.	8
III	Quantum Gates, Controlled Operations, Matrix Representation of Multi Qubit Gates, Density Matrix, Density Operator, General Properties of Density Operator, Criteria for discrimination between mixed and pure state, Quantum Circuits and its Identities, Decomposition of Quantum Gates, Single Qubit Operations, Multi Qubit	8

	Operations	
IV	QNN Models: A comparative study, Requirements for a QNN model, Concept of Quron, Implementation feasibility of Perceptron model for Boolean Reversible Functions through various Two Qubit Quantum Gates	8
V	Entangled Neural Networks (ENN), Construction of Entangled Neural Networks: Basic unit of ENN's and Structure of ENNs, Temperature adjusting problem and ENN's resolution	8
VI	Generalization Study of Quantum Neural Network: Qubit, Quantum Gates, Model Design, Data Encoding, Network Structure, Learning Algorithms, Simulating a perceptron on a quantum computer, Defining Quantum Neural Networks via Quantum Time Evolution	8
VII	Bell States, Quantum Teleportation and Superdense Coding: Principles, Proofs and Circuits, Entanglement Swapping etc, Quantum neural networks architectures for pattern classification& Clustering, pattern association and pattern mapping	8
VIII	Quantum Computing with MATLAB: Programming with QCF Library, and QETLAB 0.9 Library, Designing and Executing Quantum Circuits on Simulators such as: QCAD2000, QuIDE, Qiskit etc.	8

Suggested Readings:

1. Quantum Neural Networks by AlexenderEzhov and Dan Ventura
2. Quantum Machine Learning by Peter Wittek
3. The quest for a Quantum Neural Networks by Maria Schuld, IlyaSinayskiy, and Francesco Petruccione
4. Simulating Perceptron on a Quantum Computer by Maria Schuld, IlyaSinayskiy, and Francesco Petruccione
5. Generalization Study of Quantum Neural Network by JinZhe Jiang, Xin Zhang, Chen Li, YaQian Zhao etc.
6. Quantum Neuron: an elementary building block for machine learning on quantum computers

This course can be opted as an elective by the students of following subjects:

B.Sc. In Electronics, B.Sc. In Physics, B.Sc. In Statistics, B. Sc. Mathematics, B.Sc. In Engineering, B.Sc. Vocational, BCA, B.E./B.Tech, M.E. / M. Tech

Suggested Continuous Evaluation Methods:

2 Periodical Tests(each of 5 marks) + 10 marks for the submission of any two programs written in any programming language from the given list + 3 marks of assignment + 2 marks

Programme/Class M.Sc (Computer Science)	Year: FIFTH	Semester: X
Subject: Computer Science		
Course Code: B071003T	Course Title: Deep Learning & Pattern Recognition	
Course Outcome: -Learning of basics of Deep learning's - Pattern Recognition - image pattern		
Credit: 4		
Max. Marks: 25+75	Min. Passing Marks	

Unit	Topic	No. of lecture
I	Deep Learning Basics: Intro, History, Capabilities, The Perceptron Neural Network Learning: Back-Propagation. Autoencoders (Standard, Sparse, Denoising, Contractive, Etc), Variational Autoencoders, , Autoencoder And DBM Attention And Memory Models, Dynamic Memory Networks	8
II	Convolutional Neural Networks: Intro To Cnns, Convolution And Pooling Layers, Correlation, Filtering, Detection And Segmentation , Visualizing And Understanding , Advanced Cnns For Computer Vision. Advanced Deep Architectures: Recurrent Neural Networks (Rnns), Advanced RNN: LSTM, GRU, Deep Unsupervised Learning Deep Reinforcement Learning.	8
III	Deep Learning In NLP: Introduction To NLP And Vector Space Model Of Semantics. Word Vector Representations: Continuous Skip-Gram) Model, Continuous Bag-Of Words Model (CBOW), Glove, Evaluations And Applications In Word Similarity, Analogy Reasoning.	8
IV	Introduction: General introduction of pattern -recognition, pattern recognition tasks, difference between data and pattern, pattern classification, pattern association, pattern mapping, pattern : clustering, feature mapping, temporal pattern, pattern variability, stability plasticity dilemma, basic outline of various Pattern recognition techniques, Introduction to Statistical Pattern Recognition, Overview of Pattern Classifiers, overview of Linear Algebra: Inner product, outer product, inverses, eigen values, eigen vectors.	6
V	Bayesian decision making and Bayes Classifier: Probability: independence of events, conditional and joint probability, Random Processes: Stationary and non-stationary processes, Expectation, Autocorrelation, Cross-Correlation, spectra, Bayes Decision Theory,Bayes'theorem, Minimum-error-rate classification, Classifiers, Discriminant functions, Decision surfaces, Normal density and discriminant functions, discrete features.	7
VI	Parametric Estimation of Densities: Maximum-Likelihood estimation: Gaussian case; Maximum a Posteriori estimation; Bayesian estimation of parameters of density functions, MAP _ estimates, Bayesian Estimation examples, the exponential family of densities and ML estimates, Recursive formulation of ML and Bayesian estimates	8
VI	Unsupervised learning and _ clustering: Criterion functions for clustering; Algorithms for clustering: K-Means, Hierarchical and other methods; Cluster validation; Gaussian mixture models; Expectation-Maximization method for parameter estimation Maximum entropy estimation.	8

Program: **M.Sc. (Computer Science)**	Year: **Fifth**	Semester: **X**
Subject: Computer Science		
Course Code: B071004T	**Computer Vision**	
Course Outcomes:		
Credits:	**Core Compulsory**	
Max Marks: 25+75	**Min Passing Marks:**	
Total No. of Lectures – Tutorial – Practical (in Hours per week): L-T-P: 4:0:0		

Unit	Topics	No. of Lectures
I	**Introduction and Image Formation:** Introduction to computer vision, historic perspective and recent challenges, Geometric primitives and transformations, Photometric image formation, digital camera	08
II	**Image processing:** Point operators, Linear filtering, Nonlinear filtering Fourier transforms, Pyramids and wavelets Geometric transformations, Model fitting and optimization	08
III	**Deep Learning:** Supervised, Unsupervised learning, Deep neural network, Convolutional neural networks, transformers, and generative models	08
IV	**Recognition:** Instance recognition, Image classification Object detection Semantic segmentation	08
V	**Feature detection and matching:** Points and patches, Edges and contours, Contour tracking, Lines and vanishing points Segmentation	08
VI	**Image alignment and stitching:** Pairwise alignment Image stitching Global alignment Compositing, Applications	08
VII	**Motion estimation:** Translational alignment Parametric motion Optical flow Layered motion, Applications Structure from motion: Geometric intrinsic calibration Pose estimation	08
VIII	Depth estimation Epipolar geometry, Sparse correspondence Dense correspondence Local methods Global optimization Applications. Image-based rendering: View interpolation, Video-based rendering Applications	08

Programme/Class: M.Sc. (Computer science)	Year: Fifth	Semester: Tenth
Subject: Computer Science		
Course Code: B071009T	Principal of Software Reliability Engineering	

Course outcomes:

1. Have an understanding of the terminology, the process and the models of the software reliability engineering
2. Have learned techniques to predict and measure reliability of the software systems
3. Know how to improve reliability during the various stages of the SDLC.

Credits: 4	Specialization Group B: Elective

Max. Marks: 25+75	Min. Passing Marks:

Total No. of Lectures Tutorials Practical (in hours per week): 4-0-0

Unit	Topic	No. of Lectures
I	**Introduction:** The Need for Software Reliability, Software Reliability Engineering, Why Does Software Cost So Much? Basic Definitions and Terminologies.	7
II	**Reliability Engineering Measures:** Reliability Definitions, System Mean Time to Failure, Failure Rate Function, Reliability Function for Common Distributions, Maintainability and Availability.	8
III	**Software Engineering Assessment:** Introduction, Software Versus Hardware Reliability, Software Reliability and Testing Concepts, Software Lifecycle, Software Development Process and Its Applications, Software Verification and Validation, Data Collection and Analysis.	8
IV	**Software Reliability Modelling:** Introduction, Halstead's Software Metric, McCabe's Cyclomatic Complexity Metric, Error Seeding Models, Failure Rate Models, Curve Fitting Models, Reliability Growth Models, Non-Homogeneous Poisson Process Models, Markov Structure Models.	7
V	**NHPP Software Reliability Models:** Introduction, Parameter Estimation, NHPP Models, Applications, Imperfect Debugging Versus Perfect Debugging, A Generalized NHPP Software Reliability Model, Mean Time Between Failures for NHPP.	8
VI	**Software Cost Models:** Introduction, A Software Cost Model With Risk Factor, A Generalized Software Cost Model, A Cost Model With Multiple Failure Errors, Applications.	7
VII	**Fault- Tolerant Software:** Introduction, Basic Fault- Tolerant Software Techniques, Self- Checking Duplex Scheme, Reliability Modeling, Reduction Of Common- Cause Failures.	7
VIII	**Software Reliability Models With Environment Factors:** Introduction, Definition Of Environmental Factors, Environmental Factors Analysis, A Generalized Model With Environmental Factors, Enhanced Proportional Hazard Jelinski- Moranda, An Application With Environmental Factors.	8

Suggested Readings:

1. R.Pham: Software Reliability Springer Verlag, Singapore, 2000.
2. J.D. Musa et. al. Software Reliability Measurement, Prediction and Application, McGraw Hill, New York 1987.
3. J.D. Musa et. al. Software Reliability Engineering, TMH, New Delhi 2005.

This course can be opted as an elective by the students of following subjects:
B. Sc In Engineering, BCA, MCA, M.Sc.(IT)

Suggested Continuous Evaluation Methods: **Max. Marks: 25**

1. Assessment Type: Class Tests (Max. Marks 14)

2. Assessment Type: Quizzes/ Objective Tests / Recognition Type (such as MCQs; True or False; Matching, Classifying) /Recall Type -Filling Blanks, One word / Phrase Answers (Max Marks: 5)

3. Assessment Type: Assignments (Max Marks: 4)

4. Assessment Type: Class Interaction (Max. marks: 2)

Course prerequisites:
To study this course, a student must have had the subject Software Engineering, Software Project Management

Suggested equivalent online courses:
1. https://nptel.ac.in/courses/106/105/106105087/

Further Suggestions.
None

Programme: M.Sc. (Computer Science)	Year: Fifth	Semester: X
Subject: Computer Science		
Course Code: B071010T	IoT	
Course Outcomes:		
Credits:	Core Compulsory	
Max Marks: 25+75	Min Passing Marks:	
Total No. of Lectures – Tutorial – Practical (in Hours per week): L-T-P: 4:0:0		

Unit	Topics	No. of Lectures
I	**Introduction to Internet of Things:** Introduction – Physical Design of IoT – Logical Design of IoT – IoT Enabling Technologies – IoT & Deployment Templates. **Domain Specific IoTs:** Introduction – Home Automation – Cities – Environment – Energy – Retail – Logistics – Agriculture – Industry – Health & Life style	08
II	**IoT and M2M : Introduction:** M2M – Difference between IoT and M2M – SDN and NFV for IoT. **IoT System Management with NETCONF-YANG :** Need for IoT Systems Management – Simple Network Management Protocol (SNMP) – Network Operator Requirements – NETCONF- YANG – IoT Systems Management with NETCONF YANG	08
III	**IoT Platforms Design Methodology:** Introduction – IoT Design Methodology – Case Study on IoT System for Weather Monitoring – Motivation for using Python. **IoT Systems –Logical Design using Python:** Introduction – Installing Python – Python Data types & Data Structures – Control Flow – Functions – Modules – Packages – File Handling – Date/Time Operations – Classes – Python packages of Interest for IoT.	08

IV	**IoT Physical Devices & Endpoints:** What is an IoT Device – Exemplary Device: Raspberry Pi – About the Board – Linux on Raspberry Pi – Raspberry Pi Interfaces – Programming Raspberry Pi with Python – Other IoT devices, IoT **Physical Servers & Cloud Offerings :** Introduction to Cloud Storage Models & Communication APIs – WAMP - AutoBahn for IoT– Xively Cloud for IoT – Python Web application Framework- Django – Designing a REST ful 631 Web API – Amazon Web Services for IoT – SkynetIoT messaging platform.	**08**
V	**Case Studies Illustrating IoT Design:** Introduction – Home Automation – Cities – Environment – Agriculture – Productivity applications **Data Analytics for IoT :** Introduction – Apache Hadoop – Using Hadoop MapReduce for Batch Data Analysis – Apache Oozier – Apache Spark – Apache Storm – Using Apache Storm for Real-time Data Analysis.	**08**

This course can be opted as an elective by the students of following subjects:

B.Sc. in Electronics, B.Sc. in Physics, B.Sc. in Statistics, B.Sc. in Engineering, B.Sc. Vocational, BCA, Bachelor in Fine Arts., B.E./ B.Tech.

Suggested Continuous Evaluation Methods: **Max.Marks:25**
1. Assessment Type: Class Tests (Max. Marks14)
2. Assessment Type: Quizzes / Objective Tests/ Recognition Type (such as MCQs; True or False; Matching; Classifying) / Recall Type- Filling Blanks; One word / Phrase Answers (Max Marks:5)
3.Assessment Type: Assignments (Max Marks:4)
4.Assessment Type: Class Interaction (Max Marks:2)

Course prerequisites:
To study this course, a student must have had the subject Software Engineering

Suggested equivalent online courses

Further Suggestions:
None

Programme/Class: **M.Sc. (Computer science)**	Year: **Fifth**	Semester: **Tenth**
Subject: **Computer Science**		
Course Code: **B071015T**	Course Title: **Big Data & Data Analytics**	

Course outcomes:

CO1: To identify Big Data and its business implications.
CO2: To access and process data on distributed file system
CO3: To manage job execution in Hadoop environment
CO4: To develop Big Data solutions using Hadoop

Credits: **4**	**Specialization Group B: Elective**
Max. Marks: **25+75**	Min. Passing Marks:

Total No. of Lectures-Tutorials-Practical (in hours per week): **4-0-0**

Unit	Topic	No. of Lecture

I	**Introduction**: Types of Digital Data, Introduction to Big Data, Big Data Analytics, Big Data Platform, Challenges of Conventional Systems, Intelligent data analysis, Nature of Data, Analytic Processes and Tools, Analysis vs Reporting.	7
II	**History** of Hadoop, Apache Hadoop, Analysing Data with Hadoop, Components of Hadoop Analysing the Data with Hadoop, Scaling Out, Hadoop Streaming, Hadoop environment, Hadoop Echo System.	8
III	**Hadoop Distributed File System**: Design of HDFS, HDFS Concepts, Command Line Interface, Hadoop file system interfaces, Data flow, Data Ingest with Flume, Sqoop, Hadoop archives, Hadoop I/O: Compression, Serialization, Avro, File based Data structures, Java Interfaces to HDFS.	7
IV	**Map Reduce Application**: Developing a Map Reduce Application, How Map Reduce Works, Anatomy of a Map Reduce Job run, Failures, Job Scheduling, Shuffle and Sort, Task execution, Map Reduce Types and Formats, Map Reduce Features.	8
V	**Pig**: Introduction to PIG, Execution Modes of Pig, Comparison of Pig with Databases, Grunt, Pig Latin, User Defined Functions, Data Processing operators, Filtering, Sorting, Combining and Splitting, Modes of execution.	8
VI	**Hive**: Hive Shell, Hive Services, Hive Metastore, Comparison with Traditional Databases, Data types, Create Database, Drop database, HiveQL, Tables, Create Tables, Alter Tables, Drop Tables, Partitioning, Querying Data, Operators, User Defined Functions.	7
VII	**Hbase**: HBasics, Concepts, Clients, Example, Hbase Versus RDBMS, Shell, General Commands, API, Tables and Operations, Create and Manage Data.	7
VIII	**Big SQL**: Introduction, Preparing Big SQL Environment, Creating Directories, Getting Sample Data, Create Tables, Loading Data, Creating SQL scripts, Running Sample Query, Analysis.	8

Suggested Readings:

1. Michael Berthold, David J. Hand, "Intelligent Data Analysis", Springer, 2007.

2. Tom White "Hadoop: The Definitive Guide" Third Edition, O'reilly Media, 2012.

3. Chris Eaton, Dirk DeRoos, Tom Deutsch, George Lapis, Paul Zikopoulos, "Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data", McGrawHill Publishing, 2012.

4. Anand Rajaraman and Jeffrey David Ullman, "Mining of Massive Datasets", CUP, 2012.

5. Bill Franks, "Taming the Big Data Tidal Wave: Finding Opportunities in Huge Data Streams with Advanced Analytics", John Wiley& sons, 2012.

6. Glenn J. Myatt, "Making Sense of Data", John Wiley & Sons, 2007.

7. Pete Warden, "Big Data Glossary", O'Reilly, 2011.

8. Jiawei Han, Michelline Kamber "Data Mining Concepts and Techniques", 2nd Edition, Elsevier, Reprinted 2008.

9. Da Ruan, Guoquing Chen, Etienne E.Kerre, Geert Wets, "Intelligent Data Mining", Springer, 2007.

10. Paul Zikopoulos, Dirkde Roos, Krishnan Parasuraman, Thomas Deutsch, James Giles , David Corrigan, "Harness the Power of Big Data The IBM Big Data Platform", Tata McGraw Hill Publications, 2012.

This course can be opted as an elective by the students of following subjects:
B. Sc.In Engineering, BCA, MCA, M.Sc.(IT)

Suggested Continuous Evaluation Methods: **Max. Marks: 25**

9. Assessment Type: Class Tests (Max. Marks 14)

10. Assessment Type: Quizzes/ Objective Tests / Recognition Type (such as MCQs; True or False; Matching; Classifying) /Recall Type -Filling Blanks; One word / Phrase Answers (Max Marks: 5)

11. Assessment Type: Assignments (Max Marks: 4)

12. Assessment Type: Class Interaction (Max. marks: 2)

Course prerequisites:
To study this course, a student must have had the subject Data Structures, Python programming

Suggested equivalent online courses:

Further Suggestions:
None