Department of Higher Education
U.P. Government, Lucknow

National Education Policy-2020
Common Minimum Syllabus for all U.P. State Universities

Year wise Structure of

B.Sc. (Research) in Science

and

Master in Science (Computer Science)

Syllabus Developed by:

S. No.	Name	Designation	Department	College/ University
1	Prof. Ashutosh Gupta	Director/ Professor	School of Science	U.P.Rajarshi Tandon Open University, Prayagraj
2	Prof. Manu Pratap Singh	Professor	Dept. of Computer Science	Dr. B. R. Ambedkar University, Agra
3	Dr. Brajesh Kumar	Associate Professor	Dept. of CS & IT	MJP Rohilkhand University, Bareilly
4	Dr. Deepak Kumar Verma	Assistant Professor	Department of Computer Science and Engineering	CSJM University, Kanpur

INDEX

B.Sc. (Research) in Science AND Master in Science (Computer Science)

SUBJECT PREREQUISITES
To study the Computer Science, a student must have had the subject(s) computer science in Bachelor Level.
PROGRAMME OUTCOMES (POS): Students taking admission to B.Sc. (Research) or M.Sc. (Computer Science) program are expected to get prepared with following outcomes:

PO 1	Have an ability to identify, formulate and implement computing solutions.
PO 2	Adapt the skills to implement effective solutions for need based
PO 3	Apply the concepts of software engineering while working on big modules and or projects.
PO 4	Be familiar with current research within various fields of Computer Science and develop skills to learn new technology.
PO 5	Apply computer science theory and software development concepts to construct computing-based solutions.
PO 6	Design and develop computer programs/computer-based systems in the areas related to Neural network, compilers, image processing and Mobile applications.
PO 7	Graduates will have the skill to work on multidisciplinary tasks and will be aware of the new and emerging disciplines.

Proposed B.Sc. (Research) and M.Sc. (Computer Science) Structure (4th and 5th year)

Subject: **Computer Science**

Type of Award	Year	Sem.	Paper 1 Theory	credit	Paper 2 Theory	credit	Paper 3 Theory	credit	Paper 4 Practical	credit	Paper 5 Practical	credit	Survey/ Project/ Dissertation	credit	Total Credits
Bachelor (Research) in Science	4	VII	Compiler Design & Principles	4	Digital Image Processing	4	Software Engineering	4	Compiler Lab	4	Digital Image Processing Lab	4	Literature Survey	6	26
		VIII	Artificial Neural Network	4	Mobile Applications	4	Quantum Information and Computation	4	Artificial Neural Network Lab	4	Mobile Applications Lab	4	Research Project	6	26
Master in Science	5	IX	Information and Network Security	4	**Specialization Core Compulsory**	4	**Elective-I**	4	**Lab on Specialization Core Compulsory**	4	**Lab on Elective-I**	4	Literature Survey	6	26

		X	Parallel Computing and Algorithms	4	Elective-II	4	Elective-III	4	Lab on Elective-II	4	Lab on Elective-III	4	Dissertation	6	26

	Total Credits:	104

Criterion for Internal Assessment	
Suggested Continuous Evaluation Methods:	**Max. Marks: 25**

1. **Assessment Type: Class Tests** (Max. Marks 14)
   Suggested Usage: Include all types of questions-essay, short answer, objective; Design to test all levels of domain; Exam Blue Print be prepared to ensure inclusion of all types & levels of questions and proper sampling of content; Marking Criteria made known to students; Teacher should provide written feedback selectively and discuss answers in the class; Only Role/Code numbers , not names be written to avoid bias in marking; Display of model answer copies.
   After Completion of Unit I to Unit IV, a first class test of max. marks of 7 shall be conducted.
   After Completion of Unit V to VIII, a second class test of max. marks of 7 shall be conducted.
   If any student does not appear in any one or both class test, a makeup test shall be conducted of max. marks of 5 instead of total 14 marks.

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)
   Suggested Usage: Teachers be trained in construction, advantages, disadvantages and precautions while preparing different types of objective items; Go beyond factual information to High Order Thinking (HOT) Skills. It shall be "End of the class quiz".

3. **Assessment Type: Assignments** (Max Marks: 4)
   Suggested Usage: Some class assignments shall be given to students at the end of each Unit. Note making techniques be taught to students; Not just direct questions from notes, but application analysis and synthesis of that knowledge.

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

	Total Max. Marks: 25

Criterion for Practical Evaluation & Assessment			
Internal Assessment	Marks	External Assessment	Marks
Class	05	Viva Voce	25

Interaction				
Quiz 1	10		Execution/Demonstration	20
Quiz 2	10		Write up/theory work	20
			Practical Record File	10
Total	25			75

Criterion for Evaluation of Literature Survey/Dissertation	
External Assessment	Marks
Literature Survey Viva Voce	100
Dissertation Report and Viva Voce	100

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	Digital Image Processing	Theory	4
		B070703T	Software Engineering	Theory	4
		B070704P	Compiler Lab	Practical	4
		B070705P	Digital Image Processing Lab	Practical	4
		B070706R	Literature Survey	Research	4
4	VIII	B070801T	Artificial Neural Network	Theory	4
		B070802T	Mobile Applications	Theory	4
		B070803T	Quantum Information and Computation	Theory	4
		B070804P	Artificial Neural Network Lab	Practical	4
		B070805P	Mobile Applications Lab	Practical	4
		B070806R	Research Project	Research	4
5	IX	B070901T	Information and Network Security	Theory	4
		B070919R	Literature Survey	Research	4
		Select any one Specialization Group A OR B OR C			
		Specialization Group A: Machine Learning			
		Core Compulsory Papers			
		B070902T	Foundation on Artificial Intelligence	Core/Theory	4
		B070903P	Lab on Foundation on Artificial Intelligence	Core/Practical	4
		Select any one Elective-I Theory Paper and one Elective-I Practical Lab			
		B070904T	Machine Learning Techniques	Theory Elective	4
		B070905T	Deep Learning	Theory Elective	4
		B070906P	Lab on Machine Learning Techniques	Lab Elective	4
		B070907P	Lab on Deep Learning	Lab Elective	4
		Specialization Group B: Software Engineering			
		Core Compulsory Papers			
		B070908T	Software Project Management	Core/Theory	4
		B070909P	Lab on Software Project Management	Core/Practical	4

			Select any one Elective-I Theory Paper and one Elective-I Practical Lab		
		B070910T	Software Testing and Audit	Theory Elective	4
		B070911T	System Modeling and Simulation	Theory Elective	4
		B070912P	Lab on Software Testing and Audit	Lab Elective	4
		B070913P	Lab on System Modeling and Simulation	Lab Elective	4
		Specialization Group C: Data Science			
		Core Compulsory Papers			
		B070914T	Foundation on Data Science	Core/Theory	4
		B070915P	Lab on Foundation on Data Science	Core/Practical	4
			Select any one Elective-I Theory Paper and one Elective-I Practical Lab		
		B070904T	Machine Learning Techniques	Theory Elective	4
		B070916T	Statistics for Data Science	Theory Elective	4
		B070917P	Lab on Machine Learning Techniques	Lab Elective	4
		B070918P	Lab on Statistics for Data Science	Lab Elective	4
5	X	B071001T	Parallel Computing and Algorithms	Theory	4
		B071020R	Dissertation	Research	4
		Select one Specialization Group A OR B OR C which is chosen in IX semester			
		Specialization Group A: Machine Learning (select any Two Elective Theory Paper and its Practical Lab)			
		B071002T	Quantum Neural network	Theory Elective	4
		B071003T	Pattern Recognition	Theory Elective	4
		B071004T	Natural Language Processing	Theory Elective	4
		B071005P	Lab on Quantum Neural network	Lab Elective	4
		B071006P	Lab on Pattern Recognition	Lab Elective	4
		B071007P	Lab on Natural Language Processing	Lab Elective	4
		Specialization Group B: Software Engineering (select any Two Elective Theory Paper and Two Practical Lab)			
		B071008T	Software Architectures	Theory Elective	4
		B071009T	Software Reliability Engineering	Theory Elective	4
		B071010T	Software Security Engineering	Theory Elective	4
		B071011P	Lab on Software Architectures	Lab Elective	4
		B071012P	Lab on Software Reliability Engineering	Lab Elective	4
		B071013P	Lab on Software Security Engineering	Lab Elective	4
		Specialization Group C: Data Science (select any Two Elective Theory Paper and Two Practical Lab)			
		B071014T	Data Visualization	Theory Elective	4
		B071015T	Big Data	Theory Elective	4
		B071016T	Natural Language Processing	Theory Elective	4
		B071017P	Lab on Data Visualization	Lab Elective	4
		B071018P	Lab on Big Data	Lab Elective	4
		B071019P	Lab on Natural Language Processing	Lab Elective	4
				Total Credits	**96**

Note:

1. The student has to choose any one specialization group A, B or C in 5th year.
2. The electives for semester IX and X should be chosen from same

specialization group decided at the time of admission in 5th year
3. No electives are repeated in IX and X semester.
4. The Universities can also add new specialization groups as per their needs and available faculty.

Detailed Syllabus
for
B.Sc. (Research)
in
Science

Syllabus of VII Semester

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, 8 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),

1997. 4.   Charles N. Fischer and Ricard J. LeBlanc," Crafting a Compiler with C", Pearson Education, 1991.
This course can be opted as an elective by the students of following subjects: B.Sc. in Electronics, BCA, 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: Good understanding of computer architecture, operating system, data structure, and algorithms. Good knowledge of programming languages and ability to reason well.
Suggested equivalent online courses: 1.   https://onlinecourses.nptel.ac.in/noc21_cs07/preview 2.   https://online.stanford.edu/courses/soe-ycscs1-compilers 3.   https://www.classcentral.com/course/swayam-compiler-design-12926 4.   https://www.udemy.com/course/introduction-to-compiler-construction-and-design/
Further Suggestions: None

Edit with WPS Office

Programme/Class: **B.Sc. (Research) in Science**	Year: **Fourth**	Semester: **Seven**
Subject: **Computer Science**		
Course Code: **B070702T**	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

Suggested Readings: 1. Donald Hearn and M Pauline Baker, "Computer Graphics C Version",Prentice Hall,USA, 1997 2. Amrendra N Sinha and Arun D Udai,"Computer Graphics", Tata MCGraw Hill, India, 2008 3. Rafael C. Gonzalez, Richard E. Woods, "Digital Image Processing", Pearson, Fourth Edition, 2017 4. Anil K. Jain, "Fundamentals of Digital Image Processing", Pearson, Fourth Edition. 5. Kenneth R. Castleman, "Digital Image Processing", Pearson, 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. 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 Mathematics in class12th.**
Suggested equivalent online courses: https://onlinecourses.nptel.ac.in/noc19_ee55/preview, https://www.mooc-list.com/tags/image-processing, https://www.udemy.com/course/digital-image-processing-operations-and-applications/
Further Suggestions: **None**

Programme/Class: **B.Sc. (Research) in Science**		Year: **Fourth**	Semester: **Seven**
Subject: **Computer Science**			
Course Code: **B070703T**		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 Pedryez, "Software Engineering – An Engineering Approach",

John Wiley and Sons, New Delhi, 2000.
5. Ali Behforooz and Frederick J Hudson, "Software Engineering Fundamentals", Oxfor University Press, New Delhi, 1996.
6. Pfleeger, "Software Engineering", Pearson Education India, New Delhi, 1999.
7. Carlo Ghezzi, Mehdi Jazayari and Dino Mandrioli, "Fundamentals of Software Engineering", Prentice Hall of India, New Delhi, 1991.

This course can be opted as an elective by the students of following subjects:
B.Tech (CSE)

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:
Problem Solving using Computer, Analysis of Algorithms and Data Structures

Suggested equivalent online courses:
1. https://onlinecourses.nptel.ac.in/noc20_cs68/preview
2. https://onlinecourses.swayam2.ac.in/cec20_cs07/preview
3. https://www.coursera.org/courses?query=software%20engineering
4. https://www.edx.org/learn/software-engineering

Further Suggestions:
None

Programme/Class: Bachelor (Research) in Science	Year: Fourth	Semester: VII
Subject: Computers		
Course Code: **BT070704P**	Course Title: **Compiler Design Lab**	
Course Outcomes After the completion of the course, the student will be able to **CO 1-** Acquire the basic knowledge of implementation of lexical rules, and grammars for programming language. **CO 2-** Apply parsing techniques on given expression, based on given grammar. **CO 3-** Able to develop small tools for different compiler design concepts.		
Credits: **4**	**Core Compulsory**	
Max. Marks: **25+75**	Min. Passing Marks:	
Total No. of Lectures-Tutorials-Practical (in hours per week): L-T-P:		
Unit	Topics	No. of Lectures
I	**Introduction and Lexical Analysis** Specification & Recognition of tokens, Regular sets and expression, Finite automata, Conversion of Regular expression to Finite automata, Obtaining Regular	25

	expression from Finite Automata, LEX-compiler.	
II	**Parsing** Context Free Grammar, Derivation and Parse Tree, Basic Parsing Techniques: LL parsing, SLR Parser, LALR Parser, implementation of LR parsing tables, recursive descent parsing.	25
III	**Syntax-directed Translation** Postfix notation, Parse trees and syntax trees, L-attribute and S-attribute, three address code, quadruple, triples, Postfix notation, arithmetic expressions, procedures call, declarations and case statements.	10

Suggested Readings:

5. Aho, Sethi & Ullman, "Compilers: Principles, Techniques and Tools", 2nd Edition, Pearson Education, 2007.
6. Kenneth C. Louden," Compiler Construction", PWS Publishing Company (Cengage Learning), 1997.
7. Charles N. Fischer and Ricard J. LeBlanc," Crafting a Compiler with C", Pearson Education, 1991.
8. John Levine, Doug Brown, Tony Mason, "LEX and YACC," 2nd Edition, O'Relly, 1992.

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:

Good understanding of computer architecture, operating system, data structure, and algorithms. Good knowledge of C/C++ programming language, and ability to reason well.

Further Suggestions: Suggested Programs.

1. Provide a sample mini source language and ask the students to write a lexical analyzer in C/C++ to identify the tokens defined by the grammar.
2. Write the program to recognize the strings.
3. Write the program to detect the valid variables.
4. Write the program to identify valid operators.
5. Implement the lexical analyzer using JLex, flex or other lexical analyzer generating tools.
6. Write a C program for implementing the functionalities of predictive parser for the mini language.
7. Write a C program for constructing of LL (1) parsing.
8. *Write a C program for constructing recursive descent parsing.
9. Write a C program to implement LALR parsing.
10. Write a C program to implement operator precedence parsing.
11. Write a C program to implement Program semantic rules to calculate the expression that takes

an expression with digits, + and * and computes the value.

12. Convert the BNF rules into Yacc form and write code to generate abstract syntax tree for the mini language.
13. Write a C program to generate machine code from abstract syntax tree generated by the parser.
14. Write a LEX Program to convert the substring abc to ABC from the given input string.
15. Write a LEX program to find out total number of vowels and consonants from the given input sting.

Programme/Class: **B.Sc. (Research) in Science**	Year: **Fourth**	Semester: **Seven**
Subject: **Computer Science**		
Course Code: **B070705P**	Course Title: **Digital Image Processing Lab**	
Course outcomes: CO1: Review the fundamental concepts of a digital image processing system. CO2 : Analyze images in the frequency domain using various transforms. CO3 : Evaluate the techniques for image enhancement and image restoration. CO4 : Categorize various compression techniques. CO5: Interpret Image compression standards. CO6 : Interpret image segmentation and representation techniques.		
Credits**: 4**	Max. Marks: **25+75**	Min. Passing Marks:
Total No. of Lectures-Tutorials-Practical (in hours per week): **0-0-8**		
Suggested Readings: 1. Rafael C. Gonzalez, Richard E. Woods, "Digital Image Processing", Pearson, Fourth Edition, 2017		

2. Anil K. Jain, "Fundamentals of Digital Image Processing", Pearson, Fourth Edition.

3. Kenneth R. Castleman, "Digital Image Processing", Pearson, 2006.

4. Rafael C. Gonzalez, Richard E. Woods, Steven Eddins, "Digital Image Processing using MATLAB" Pearson Education, Inc., 2011.

Programs:

1. Write a program to rotate a circle (alternatively inside and outside) around the circumference of another circle.
2. Write a program to draw a cube using in build library function and perform 3D transformations i) Translations in x, y, z directions ii) Rotation by angle 450 about z axis, rotation by 600 about y-axis in succession. iii) Scaling in x-direction by a factor of 2, scaling in y- direction by a factor of 3.
3. Write a program to implement line clipping (Cohen Sutherland algorithm).
4. Write a program to draw balloons using in build graphics function and translate it from bottom left corner to right top corner of screen.
5. Write a program to draw a circle using midpoint algorithm. Write a program to draw a circle using Bresenham's algorithm.
6. Write a program to display grayscale image using read and write operation.
7. Write a program to apply linear and power log transformation functions on an image.
8. Write a program to find histogram value and display histogram of a grayscale and color image.
9. Write a program to find the negative of an image.
10. Write a program to apply Linear Spatial Filters on images.
11. Write a vision program to determine the edge detection of an image using different operators.
12. Write a program to discretize an image using Fourier transformation.
13. Write a program to eliminate the high frequency components of an image.
14. Write a program for a color image to perform read and write operation.
15. Write a program to display an image in Red, GrayScale, Green, and Blue.
16. Write a program that performs discrete wavelet transform on image.
17. Write a program for segmentation of an image using watershed transforms.
18. Write a program to show Image Enhancement using various filters Sobel, Previt, Laplacian.
19. Write a program to observe the effect of median filter on an image corrupted by salt and pepper effect.
20. Write a program to perform Dilation and Erosion operations of morphology on images.
21. Write a program to blur an image using Gaussian method.

Syllabus of VIII Semester

Programme/Class: Bachelor(Research) in Science	Year: **Fourth**	Semester: Eighth
Subject: **Computer Science**		
Course Code: **B070801T**	Course Title: **Artificial Neural Network**	

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: **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. Maximiliano
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/

Further Suggestions: **None**	

Programme/Class: **B.Sc. (Research) in Science**	Year: **Fourth**	Semester: **Eighth**
Subject: **Computer Science**		
Course Code: B070803T	Course Title: **Quantum Information Computing**	

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's-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/Class: **B.Sc. (Research) in Science**	Year: **Fourth**	Semester: **Eighth**
Subject: **Computer Science**		
Course Code: **B070804P**	Course Title: **Artificial Neural Network Lab**	

Course outcomes:
After Completing the course students will be able to
CO1: Understand the characteristics and types of artificial neural network and remember working of biological Neuron and Artificial Neural Network.
CO2: Apply learning algorithms on perceptron and apply back propagation learning on Neural Network.
CO3: Implement different types of correlators.
CO4: Implement Logic gates..

Credits: **4**	Max. Marks: **25+75**	Min. Passing Marks:
Total No. of Lectures-Tutorials-Practical (in hours per week): **0-0-8**		

Suggested Readings:
1. Rafael C. Gonzalez, Richard E. Woods, "Digital Image Processing", Pearson, Fourth Edition, 2017
2. Anil K. Jain, "Fundamentals of Digital Image Processing", Pearson, Fourth Edition.
3. Kenneth R. Castleman, "Digital Image Processing", Pearson, 2006.
4. Rafael C. Gonzalez, Richard E. Woods, Steven Eddins, "Digital Image Processing using MATLAB" Pearson Education, Inc., 2011.

List of experiments: (Use Python OR MATLAB)

1 Create a perceptron with appropriate no. of inputs and outputs. Train it using fixed increment learning algorithm until no change in weights is required. Output the final weights.

2 Create a simple ADALINE network with appropriate no. of input and output nodes. Train it using delta learning rule until no change in weights is required. Output the final weights.

3 Train the autocorrelator by given patterns: A1=(-1,1,-1,1), A2=(1,1,1,-1), A3=(-1, -1, -1,1). Test it using patterns: Ax=(-1,1,-1,1), Ay=(1,1,1,1), Az=(-1,-1,-1,-1).

4 Train the hetrocorrelator using multiple training encoding strategy for given patterns: A1=(000111001) B1=(010000111), A2=(111001110) B2=(100000001), A3=(110110101) B3(101001010). Test it using pattern A2.

5. To Write a program to implement Perceptron.

6. To write a program to implement AND OR gates using Perceptron.

7. Build an Artificial Neural Network by implementing the Backpropagation algorithm and test the same using appropriate data sets.

8. Implementation of logic gate model (AND,OR,NOT,NAND,NOR) using McCulloh− Pitts model.

9. Implementation of Unsupervised Learning Algorithm.

10. Implement the Hopfield neural network for pattern storage and recalling.

11. Implement the Boltzmann machine for pattern storage and recalling using simulated annealing.

12. Implement the linear associator for pattern recalling and encoding.

13. Implement the RBF network for regularization and approximation.

14. Implement the pattern clustering for the given set of patterns using competitive learning.

15. Implement the SOM for vector quantization and data compression.

16. Implement the SOM for feature mapping.

17. Implement neural network for pattern classification of given sample patterns.

Programme/Class: **B.Sc. (Research) in Science**	Year: **Fourth**	Semester: **Eighth**
Subject: **Computer Science**		
Course Code: **B070805P**	Course Title: **Mobile Applications Lab**	

Course outcomes:

The student should be able to:
1. Install Andriod Environment.
2. Understand resources, layouts, and intents
3. Develop small application on android platform.
4. Develop small application using SQLite

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

Total No. of Lectures-Tutorials-Practical (in hours per week): **0-0-8**

Suggested Readings:

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

Programs:

Mobile Applications Lab

1. Installing Android Environment
2. Create Hello World Application
3. Sample Application about Android Resources
4. Sample Application about Layouts
5. Sample Application about Intents
6. Sample Application about user interfaces
7. Sample Application about Animations
8. Sample Application about Android Data
9. Sample Application about SQLite I
10. Sample Application about SQLite II
11. Make a Project based on above concepts

Detailed Syllabus
for
M.Sc.
in
Computer Science

Programme/Class: M.Sc. (Computer Science)	Year: Fifth	Semester: Ninth
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 + 3 marks of assignment + 2 marks of attendance.

Course prerequisites:

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

Edit with WPS Office

Programme/Class: **M.Sc. (Computer science)**	Year: **Fifth**	Semester: **Tenth**
Subject: **Computer Science**		
Course Code: **B071001T**	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-performance-computing

Further Suggestions:
None

Specialization Group A: Machine Learning

Paper Code	Paper Title	Core/Elective	Theory/ Practical	Credit	Semester
B070902T	Foundation on Artificial Intelligence	Core	Theory	4	9
B070903P	Lab on Foundation on Artificial Intelligence	Core	Practical	4	9
Choose Electives I from the list					
B070904T	Machine Learning Techniques	Theory Elective	Theory	4	9
B070905T	Deep Learning	Theory Elective	Theory	4	9
B070906P	Lab on Machine Learning Techniques	Lab Elective	Practical	4	9
B070907P	Lab on Deep Learning	Lab Elective	Practical	4	9
Choose Electives II, III from the list					
B071002T	Quantum Neural network	Theory Elective	Theory	4	10
B071003T	Pattern Recognition	Theory Elective	Theory	4	10
B071004T	Natural Language Processing	Theory Elective	Theory	4	10
B071005P	Lab on Quantum Neural network	Lab Elective	Practical	4	10
B071006P	Lab on Pattern Recognition	Lab Elective	Practical	4	10
B071007P	Lab on Natural Language Processing	Lab Elective	Practical	4	10

Programme: **Master in Science (Computer Science)**	Year: **Fifth**	Semester: **IX**
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

42

	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.1, Backward Reasoning Method, Resolution, Rules Of Inference, Modus Pones, 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.sanfoundry.com/artificial-intelligence-mcqs-inductive-logic-unification-lifting-1/

Programme: **Master in Science (Computer Science)**	Year: **Fifth**	Semester: **IX**
Subject: **Computer Science**		
Course Code: **B070903P**	Course Title: **Lab on Artificial Intelligence**	
Course outcomes: Upon the completion of the course, the student will be able implement 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**: 4**	Max. Marks: **25+75**	Min. Passing Marks:
Total No. of Lectures-Tutorials-Practical (in hours per week): **0-0-8**		
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		

<div align="center">

Lab on Artificial Intelligence

</div>

List of Practical in AI with python:

1. Write a program to implement Tic-Tac-Toe game problem.

2. Write a program to implement BFS (for 8 puzzle problem or Water Jug problem or any AI search problem) .

3. Write a program to implement DFS (for 8 puzzle problem or Water Jug problem or any AI search problem) .

4. Write a program to implement Single Player Game (Using Heuristic Function)

5. Write a program to Implement A* Algorithm.

6. Write a program to solve N-Queens problem.

7. Write a program to solve travelling salesman problem.

8. Write a program to implement hill climbing algorithm.

9. Write a program to implement Maxmin algorithm for two player's game.

10. Write a program to implement unification process and resolution process.

Programme/Class: **Master in Science (Computer Science)**	Year: **Fifth**	Semester: **IX**
Subject: **Computer Science**		
Course Code: **B070904T**	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. Logisitic 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 class12[th].

Suggested equivalent online courses:

Further Suggestions:
Programs:

Edit with WPS Office

Programme: **Master in Science (Computer Science)**	Year: **Fifth**	Semester: **IX**
Subject: **Computer Science**		
Course Code: **B070906P**	Course Title: **Lab on 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**	Max. Marks: **25+75**	Min. Passing Marks:
Total No. of Lectures-Tutorials-Practical (in hours per week): **0-0-8**		
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

Lab on on Machine Learning Techniques
List of Practical in AI with python:

1. Write a program to implement k-Nearest Neighbour algorithm to classify the iris data set. Print both correct and wrong predictions. Python ML library classes can be used for this problem.
2. Write a program to construct a Bayesian network considering medical data. Use this model to demonstrate the diagnosis of heart patients using standard Heart Disease Data Set. You can use Python ML library classes/API.
3. Assuming a set of documents that need to be classified, use the naïve Bayesian Classifier model to perform this task. Calculate the accuracy, precision, and recall for your data set.
4. Implement and demonstrate the FIND-S algorithm for finding the most specific hypothesis based on a given set of training data samples.
5. Write a program to demonstrate the working of the decision tree based ID3 algorithm. Use an appropriate data set for building the decision tree and apply this knowledge to classify a new sample.
6. Implement the non-parametric Locally Weighted Regression algorithm in order to fit data points. Select appropriate data set for your experiment and draw graphs.
7. Implement the Viterbi Algorithm in Hidden Markov Model.
8. Implements the Forward-backward EM algorithm for training the parameters of HMM's
9. Implement the HMM for pattern classification of English vowels.

Programme: **Master in Science (Computer Science)**	Year: **Fifth**	Semester: **IX**
Subject: **Computer Science**		
Course Code: **B070905T**	Course Title: **Deep Learning**	

Course outcomes:After the completion of the Deep Learning course, the student would have the knowledge and understanding of the fundamentals of deep learning., he will know the main techniques in deep learning and would be able to design and implement deep neural network systems. The student will be able to identify new application requirements in the field of computer vision. He would be able to identify reasonable work goals and estimate the resources required to achieve the objectives. He will be able to structure and prepare scientific and technical documentation describing project activities. Through his learning skills, he will be able to autonomously extend the knowledge acquired during the study course by reading and understanding scientific and technical documentation.

Credits: **4**		Specialization 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	Deep Learning Basics: Intro, History, Capabilities, The Perceptron Neural Network Learning: Back-Propagation.		6
II	Autoencoders (Standard, Sparse, Denoising, Contractive, Etc), VariationalAutoencoders, , Autoencoder And DBM Attention And Memory Models, Dynamic Memory Networks		7
III	Convolutional Neural Networks: Intro To Cnns, Convolution And Pooling Layers, Correlation, Filtering, Detection And Segmentation , Visualizing And Understanding , Advanced Cnns For Computer Vision.		8
IV	Advanced Deep Architectures: Recurrent Neural Networks (Rnns), Advanced RNN: LSTM, GRU, Deep Unsupervised Learning Deep Reinforcement Learning.		8
V	Deep Learning In NLP: Introduction To NLP And Vector Space Model Of Semantics.		8
VI	Word Vector Representations: Continuous Skip-Gram Model, Continuous Bag-Of Words Model (CBOW), Glove, Evaluations And Applications In Word Similarity, Analogy Reasoning .		8
VII	Generative Adversarial Networks (GANs), Advanced GANs, Encoder Decoder Architectures,		8
VIII	Introduction Of Tools: Tensorflow ,Pytorch. Case Study: Computer Vision , Natural Language Processing (NLP), Sequence Modeling , Natural / Biological Signals, Face Recognition		7

Suggested Readings:

1. Bengio, Yoshua, Ian J. Goodfellow, and Aaron Courville. "Deep learning." MIT Press,2015.
2. Nikhil Buduma ,"Fundamentals of Deep Learning", SPD
3. Dr. Pablo Rivas, "Deep Learning for Beginners", 2020
4. Nikhil Singh Paras Ahuja, "Fundamentals of Deep Learning and Computer Vision", BPB Publications.
   1. https://blog.algorithmia.com/introduction-natural-language-processingnlp/
   2. https://www.udacity.com/course/natural-language-processingnanodegree--nd892
   3. https://www.coursera.org/learn/language-processing
   4. https://towardsdatascience.com/a-practitioners-guide-to-naturallanguage-processing-part-i-processing-understanding-text-9f4abfd13e72
   5. https://www.edx.org/course/natural-language-processing

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, 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:
Basic knowledge in the following topics is required: • Linear algebra • Calculus • Statistics • Basic programming in Python • Machine learning

Suggested equivalent online courses:
http://www.cs.bilkent.edu.tr/~gcinbis/courses/Spring17/CS559
https://www.coursera.org/specializations/deep-learning

Further Suggestions:
https://www.greatlearning.in/academy/learn-for-free/courses/introduction-to-neural-networks-and-deep-learning
https://www.classcentral.com/subject/deep-learning

Programme: **Master in Science (Computer**	Year: **Fifth**	Semester: **IX**

Science)		
Subject: **Computer Science**		
Course Code: **B070907P**	Course Title: **Lab on Deep Learning**	

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

Total No. of Lectures-Tutorials-Practical (in hours per week): **0-0-8**

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

Lab on Deep Learning
List of Practical in Deep Learning with python:

1. Explore image classification with two neural network architectures: multi-layer perceptrons (MLP) and convolutional neural networks (CNN).

2. Explore the long-term dependency modelling capabilities of Recurrent Neural Networks (RNNs) and Long Short-Term Networks (LSTMs)

3. Explore deep generative modelling with deep generative models, and Generative Adversarial Networks.

4. Train a neural network from scratch to classify data using TensorFlow 2, and how to use the weights of an already trained model to achieve classification to another set of data.

5. implement the autoencoder, stacking an encoder and decoder using TensorFlow 2, and depict reconstructed output images by the autoencoder model

6. Implement and apply a Softmax classifier

7. Implement Batch Normalization and Layer Normalization for training deep networks.

8. Train and implement a Generative Adversarial Network (GAN) to produce images that resemble samples from a dataset.

9. Implement the changes needed in backward propagation to take into account regularization.

10. Implement the LeNet-5 model in TensorFlow using HW2_3a_template.ipynb

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

of attendance.	

Course prerequisites:	
Students those have already studied the paper of Quantum Computation can offer this paper. Informally, student should be familiar with calculus and linear algebra, and know some probability and discrete math. Knowledge of quantum computation or quantum information processing is prerequisite. Similarly the knowledge of Artificial neural networks is also required.	
Suggested equivalent online courses:	
Learning website: https://www.mooc-list.com/course/quantum-machine-learning-edx, https://ai.googleblog.com/2018/12/exploring-quantum-neural-networks.html	

Programme/Class: Master in Science	Year: Fifth	Semester: X
Subject: **Computer Science**		
Course Code: **B071003T**	Course Title: **Pattern Recognition**	
Course outcomes: Students will learn the fundamentals of pattern recognition and its relevance to classical and modern problems. Student will also understand the concepts, theory and computational algorithms needed for several real world recognition tasks such as text, speech, characters, objects etc. Simulate and understand how machine will have power to accomplish these tasks and can aim at developing several examples based learning tasks in several domains ranging from medical, economical, engineering to industrial needs. After taking the course, the student should have a clear understanding design and implementation of a pattern recognition system. The student should also have some exposure to the theoretical issues involved in pattern recognition system design such as the curse of dimensionality. Finally, the student will have a clear working knowledge of implementing pattern recognition techniques and the scientific Python computing environment.		
Credits: **4**	Group A: 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**: 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

II	**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
III	**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 MLestimates, Recursive formulation of ML and Bayesian estimates.	8
IV	**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	7
V	**Sequential Pattern Recognition**: Hidden Markov Models (HMMs); Discrete HMMs; Continuous HMMs, Convergence of expectation-maximization algorithm, overview of Nonparametric density estimation,Nonparametric techniques for density estimation,Parzen-window method; K-Nearest Neighbour method	8
VI	**Dimensionality reduction:**Fisher discriminant analysis; Principal component analysis; Factor Analysis,Linear discriminant functions: Gradient descent procedures; Perceptron; Support vector machines,Linear Least Squares Regression, AdaLinE and LMS algorithm; General non-liner least-squares regression, Logistic Regression, Statistics of least squares method; Regularized Least Squares.	8
VII	**Non-metric methods for pattern classification**: Non-numeric data or nominal data; Decision trees: CART, Linear Discriminant functions for multi-class case; multi-class logistic regression, - Feed-forward networks for Classification and Regression, Radial Basis Function	8

	Networks; Gaussian RBF networks	
VIII	**Support Vector Machines and Kernel based methods**: Support Vector Machines ,Introduction, obtaining the optimal hyper plane, SVM formulation with slack variables; nonlinear SVM classifiers Kernel Functions for nonlinear SVMs; Mercer and positive definite Kernels, Support Vector Regression and ε-insensitive Loss function, examples of SVM learning, Overview of SMO and other algorithms for SVM; v -SVM and v-SVR;SVM as a risk minimizer, Positive Definite Kernels; RKHS; Representer Theorem.	8

Suggested Readings:

1. R.O.Duda,P.E.Hart and D.G.Stork,Pattern Classification, John Wiley, 2002.

2. C.M.Bishop, Neural Networks and Pattern Recognition, Oxford University Press (Indian Edition), 2003

3. Bishop, C. M. Pattern Recognition and Machine Learning. Springer. 2007.

4. Marsland, S. Machine Learning: An Algorithmic Perspective. CRC Press. 2009. (Also uses Python.) Theodoridis, S. and Koutroumbas, K. Pattern Recognition. Edition 4. Academic Press, 2008.

5. Russell, S. and Norvig, N. Artificial Intelligence: A Modern Approach. Prentice Hall Series in Artificial Intelligence. 2003.

6. Bishop, C. M. Neural Networks for Pattern Recognition. Oxford University Press. 1995.

7. Hastie, T., Tibshirani, R. and Friedman, J. The Elements of Statistical Learning. Springer. 2001.

8. Koller, D. and Friedman, N. Probabilistic Graphical Models. MIT Press. 2009.

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, 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 up to graduation level and must have the good knowledge to liner algebra, probability, statistics and mathematics. It is assumed the students have a working knowledge of calculus, linear algebra,

and probability theory. It is also assumed the students have some experience programming in a scientific computing environment.

Suggested equivalent online courses:
- https://www.classcentral.com/course/swayam-pattern-recognition-and-application-14228
- https://onlinecourses.nptel.ac.in/noc19_ee56
- https://etu.ru/en/university/news/the-online-course-on-pattern-recognition-and-machine-learning

Programme/Class: **M.Sc. (Computer science)**	Year: **Fifth**	Semester: **Tenth**
Subject: **Computer Science**		
Course Code: **B071004T**	Course Title: **Natural Language Processing**	

Course outcomes:
CO1: Understand the mathematical and linguistic foundations for natural language processing.
CO2: Understand approaches to syntax and semantics in NLP.
CO 3: Understand approaches to syntax and semantics in NLP.
CO 4: Understand approaches to discourse, generation, dialogue and summarization within NLP.
CO 5:Understand current methods for statistical approaches to machine translation.
CO 6: Understand machine learning techniques used in NLP

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	**Natural language and Formal language**, NLP tasks in syntax, semantics, and pragmatics, Applications such as information extraction,The problem of ambiguity, The role	7

59

Edit with WPS Office

	of machine learning in NLP, ArgMax Computation.	
II	**WSD**: WordNet, Wordnet; Application in Query Expansion, Wiktionary; semantic relatedness, WordNet Similarity, N-grams and language models, Corpora, Unigram, Bigram, and Trigram models, Usage of N-grams, N-grams Training & Testing.	8
III	**N-grams** (cont.), Counting, Probability, Perplexity, Entropy, Smoothing techniques, Backoff methods, Class-based models. Part of Speech (POS) Tagging, POS Tagger, Chunking.	7
IV	**Parsing Algorithms**, Evidence for Deeper Structure; Top Down Parsing Algorithms, Noun Structure, Non-noun Structure, Probabilistic parsing; sequence labeling, PCFG, Probabilistic parsing: Training issues, Probabilistic parsing; inside-outside probabilities.	8
V	**Text Clustering**, Distributional Semantics, Morphology, Graphical Models for Sequence Labelling in NLP, Phonetics.	8
VI	Consonants, Vowels, Phonology, HMM and Viterbi, Forward Backward probability, Opinions on the Web, Machine Translation, Text Entailment.	7
VII	**Sentiment Analysis**, Semantic Analysis, Text coherence and discourse structure, Information extraction, Information retreival, Pronounse, Reference resolution.	7
VIII	**Precision**, Recall, F-score, Map, Semantic Relations; UNL; Towards Dependency Parsing, Universal Networking Language, Semantic Role Extraction, Baum Welch Algorithm; HMM training.	8

Suggested Readings:
1. Allen, James, Natural Language Understanding, Second Edition, Benjamin/Cumming, 1995.
2. Charniack, Eugene, Statistical Language Learning, MIT Press, 1993.
3. Jurafsky, Dan and Martin, James, Speech and Language Processing, Second Edition, Prentice Hall, 2008.
4. Manning, Christopher and Heinrich, Schutze, Foundations of Statistical Natural Language Processing, MIT Press, 1999.
1. https://blog.algorithmia.com/introduction-natural-language-processingnlp/
2. https://www.udacity.com/course/natural-language-processingnanodegree--nd892
3. https://www.coursera.org/learn/language-processing
4. https://towardsdatascience.com/a-practitioners-guide-to-naturallanguage-processing-part-i-processing-understanding-text-9f4abfd13e72
5. https://www.edx.org/course/natural-language-processing

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, Python programming
Suggested equivalent online courses:
Further Suggestions: **None**

Programme/Class: **M.Sc. (Computer science)**	Year: **Fifth**	Semester: **X**
Subject: **Computer Science**		
Course Code: **B071005P**	Course Title: **Lab on 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**	Max. Marks: **25+75**	Min. Passing Marks:
Total No. of Lectures-Tutorials-Practical (in hours per week): **0-0-8**		

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

List of experiments using MATLAB library

2. Implement the Simple model of quantum neural networks and show the stable states.
11 Implement the Quantum circuit for preparing the Bell state.
12 Implement the different quantum gates and show the outcomes.
13 Implement the measurement of stable states for feed forward quantum neural network and feedback quantum neural networks.
14 Implement the Quantum Perceptron neural network Architecture and learning rule for the pattern classification.
15 Use the Grover iterative approach to obtain the unitary operator for quantum neural network to perform the pattern association task.
16 Implement the Backpropagation learning rule for quantum neural network to perform the pattern mapping. .

Programme: **Master in Science (Computer Science)**	Year: **Fifth**	Semester: **X**
Subject: **Computer Science**		
Course Code: **B071006P**	Course Title: **Lab on Pattern Recognition**	
Course outcomes: After taking the course, the student should have a clear understanding design and implementation of a pattern recognition system. The student should also have some exposure to the theoretical issues involved in pattern recognition system design such as the curse of dimensionality. Finally, the student will have a clear working knowledge of implementing pattern recognition techniques and the scientific Python computing environment		
Credits: **4**	Max. Marks: **25+75**	Min. Passing Marks:
Total No. of Lectures-Tutorials-Practical (in hours per week): **0-0-8**		
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

Lab on Pattern Recognition

List of Practical in Pattern Recognition with python:
1. Implement the non-parametric Locally Weighted Regression algorithm in order to fit data points. Select appropriate data set for your experiment and draw graphs.
2. Write a program to construct a Bayesian network considering medical data. Use this model to demonstrate the diagnosis of heart patients using standard Heart Disease Data Set. You can use Python ML library classes/API.
3. Assuming a set of documents that need to be classified, use the naïve Bayesian Classifier model to perform this task. Calculate the accuracy, precision, and recall for your data set.
4. Write a program to construct a Bayesian network considering medical data. Use this model to demonstrate the diagnosis of heart patients using standard Heart Disease Data Set.
5. Write a program of liner regression for the classification problem.
6. Evaluate the classifier on the test set by calculating the confusion matrix and the overall accuracy.
7. Generating features for two classes and analyzing them.
8. Building classifiers for two classes using Bay's rule. •
9. Building classifiers for multiple classes using Bay's rule.
10. Building classifiers for two classes using linear classifier.
11. Building classifiers for two classes using SVM
12. Implementation of clustering of patterns
13. Implement Gradient descent algorithm for pattern mapping.
14. Implement K-means algorithm for clustering.

Programme/Class: **M.Sc. (Computer science)**	Year: **Fifth**	Semester: **Tenth**
Subject: **Computer Science**		
Course Code: **B071007P**	Course Title: **Lab on Natural Language Processing**	

Course outcomes:
CO1: Understand the mathematical and linguistic foundations for natural language processing.
CO2: Understand approaches to syntax and semantics in NLP.
CO34: Understand approaches to discourse, generation, dialogue and summarization within NLP.
CO 4:Understand current methods for statistical approaches to machine translation.
CO 5: Understand machine learning techniques used in NLP.

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

Total No. of Lectures-Tutorials-Practical (in hours per week): **0-0-8**

Suggested Readings: 1. Jay Liebowitz, —Big Data And Business Analytics Laboratory, CRC Press.

Virtual Lab on Natural Language Processing https://nlp-iiith.vlabs.ac.in/ **List of experiments available at virtual Lab are given below:** 1. Word Analysis 2. Word Generation 3. Morphology 4. N-Grams 5. N-Grams Smoothing 6. POS Tagging: Hidden Markov Model 7. POS Tagging: Viterbi Decoding 8. Building POS Tagger 9. Chunking 10. Building Chunker **NOTE: At least 8 Experiments from the above list must be done in the semester.**

Specialization Group B: Software Engineering

Paper Code	Paper Title	Core/Elective	Theory/ Practical	Credit	
B070908T	Software Project Management	Core	Theory	4	9
B070909P	Lab on Software Project Management	Core	Practical	4	9
Choose Electives I from the list					
B070910T	Software Testing and Audit	Theory Elective	Theory	4	9
B070911T	System Modeling and Simulation	Theory Elective	Theory	4	9
B070912P	Lab on Software Testing and Audit	Lab Elective	Practical	4	9
B070913P	Lab on System Modeling and Simulation	Lab Elective	Practical	4	9
Choose Electives II, III from the list					
B071008T	Software Architectures	Theory Elective	Theory	4	10
B071009T	Software Reliability Engineering	Theory Elective	Theory	4	10
B071010T	Software Security Engineering	Theory Elective	Theory	4	10
B071011P	Lab on Software Architectures	Lab Elective	Practical	4	10
B071012P	Lab on Software Reliability Engineering	Lab Elective	Practical	4	10
B071013P	Lab on Software Security Engineering	Lab Elective	Practical	4	10

Programme/Class: **M.Sc. (Computer science)**	Year: **Fifth**	Semester: **Ninth**

Subject: **Computer Science**	
Course Code: **B070908T**	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.Huges and M.Cotterell- 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

Edit with WPS Office

Programme/Class: **M.Sc. (Computer science)**	Year: **Fifth**	Semester: **Ninth**
Subject: **Computer Science**
Course Code: **B070909P**		Course Title: **Lab on Software Project Management**

Course outcomes:

The student should be able to understand the software life cycle phases (Project Management, Requirements Engineering, Software Design and Testing)

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

Total No. of Lectures-Tutorials-Practical (in hours per week): **0-0-8**

Suggested Readings:
1. B.Huges and M.Cotterell- Software Project Management 3rd Edn, TMH, New Delhi, 2004.
2. P.Jolote- Software Project Management in Practice, Pearson Education, New Delhi, 2002.

Programs:

Lab on Software Project Management
1. Preparation of SRS for any domain problem. (eg. Banking, Library, Healthcare etc).
2. Preparation of Requirements Management Plan and Software Project Management plan for the selected project.
3. Analyze the risk related to the project and prepare Risk Management Plan for the project.
4. Draw ER diagram, Data Flow Diagram, Sequence, Collaboration, Activity & State Transition diagrams for the project using appropriate tools.
5. Preparation of Test Plan and Develop Test Case Hierarchy.
6. Perform various types of testing using appropirate tools.
    (a)   Unit Testing
    (b)   Integration Testing
    (c)   Validation Testing
    (d)   Performance Testing
7. Generation of Test Cases and Test Documentation for the selected project domain.
8. Preparation of Software Configuration Management Plan.
9. Preparation of Time−Line Chart and project table using PERT or CPM project scheduling methods.

Programme/Class: M.Sc. (Computer science)	Year: Fifth	Semester: Ninth
Subject: Computer Science		
Course Code: B070910T	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.C. 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/Class: **M.Sc.** **(Computer science)**	Year: **Fifth**	Semester: **Ninth**
Subject: **Computer Science**		
Course Code: **B070912P**	Course Title: **Lab on Software Testing and Audit**	

Course outcomes:
1. To Gain Knowledge in the Test Environment
2. Ability to plan tests
3. Ability to execute tests, design test cases, use test tools, etc
4. Ability to develop testing status reports

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

Total No. of Lectures-Tutorials-Practical (in hours per week): **0-0-8**

Suggested Readings:
1. Testing in 30+ Open Source Tools, Rahul Shende, Shroff Publishers & Distributor Pvt. Ltd, ISBN 13: 9789350231005 (page numbers from 15 to 117)
2. http://seleniumhq.org/
3. http://sourceforge.net/projects/sahi/
4. http://testng.org/doc/index.html

Programs:

Lab on Software Testing and Audit

Tool Required: **Smartbear or any other test tool**
1. Write programs in C Language to demonstrate the working of the following constructs:
   (a) do….while
   (b) while…. Do
   (c) ff… else
   (d) for
2. Take any system (e.g. ATM system) and study its system specifications and report the various bugs.
3. Write test cases for any known application (e.g. banking system)
4. Study of any web testing tool (e.g. Selenium)
5. Study of any bug tracking tool (e.g. Bugzilla, bugbit)
6. Study of any test management tool (e.g. Test Director)
7. Study of any open source testing tool (e.g. Test Link)
8. To determine the nature of roots of a quadratic equations, its input is triple of +ve integers (say x,y,z) and values may be from interval[1,100] the program output may have one of the following:-
   [Not a Quadratic equations, Real roots, Imaginary roots, Equal roots] Perform BVA.
9. To determine the type of triangle. Its input is triple of +ve integers (say x,y,z) and the values may be from interval[1,100].The program output may be one of the following [Scalene, Isosceles, Equilateral, Not a Triangle].Perform BVA
10. Perform robust case testing on Problem No. 8.
11. Perform robust case testing on Problem No. 9.
12. Create a test plan document for any application (e.g. Library Management System)

71

13. Study of any testing Tool (e.g. Win Runner)
14. Test Reporting Experiment: Study of any open source testing tool (Web Performance Analyzer/O STA)
15. Write test cases to validate a mobile number using one time pin identification(OTP)
16. Write and test a program to update 10 student records into table into Excel file.
17. Using Selenium IDE, Write a test suite containing minimum 4 test cases.
18. Conduct a test suite for any two web sites.
19. Write and Test a program to find out list of employees having salary greater than Rs 50,000 and age between 30 to 40 years.
20. Understanding Test Automation. Using Selenium write a simple test script to validate each field of the registration page (Eg: Facebook Registration Page)
21. Install Selenium server and demonstrate it using a script in Java/PHP.
22. Write and test a program to select the number of students who have scored more
    than 60 in any one subject (or all subjects).
23. Write and test a program to count number of items present on a desktop.

NOTE: At least 12 Experiments out of the list must be done in the semester.

Programme/Class: **M.Sc. (Computer science)**	Year: **Fifth**	Semester: **Ninth**
Subject: **Computer Science**		
Course Code: **B070911T**	Course Title: **System Modeling and Simulation**	

Course outcomes:
1. Understand the concept of simulation, the fundamental logic, structure, Components, types of simulation models and discrete event simulation.
2. Develop solutions for application problems using manual simulation and Time Advance algorithm on discrete event simulation.
3. Understand the concepts of Statistical models and queuing models.
4. Apply acceptance rejection technique and inverse transform technique to generate Random Variates and Random numbers using LCM.
5. Understand the useful model of input data, absolute performance and estimation with respect to output analysis.
6. Understand the model building, verification, calibration, validation of models and optimization.

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:** When simulation is the appropriate tool and when it is not appropriate; Advantages and disadvantages of Simulation; Areas of application; Systems and system environment; Components of a system; Discrete and continuous systems; Model of a system; Types of Models; Discrete-Event System Simulation; Steps in a Simulation Study. The basics of Spreadsheet simulation, Simulation example: Simulation of queuing systems in a spreadsheet.	8

73

II	**General Principles, Simulation Software:** Concepts in Discrete-Event Simulation: The Event-Scheduling / Time-Advance Algorithm, World Views, Manual simulation Using Event Scheduling; List processing. Simulation in Java; Simulation in GPSS	7
III	**Statistical Models in Simulation:** Review of terminology and concepts; Useful statistical models; Discrete distributions; Continuous distributions; Poisson process; Empirical distributions.	7
IV	**Queuing Models:** Characteristics of queuing systems; Queuing notation; Long-run measures of performance of queuing systems; Steady-state behavior of M/G/1 queue; Networks of queues; Rough-cut modeling: An illustration..	8
V	**Random-Number Generation, Random-Variate Generation:** Properties of random numbers; Generation of pseudo-random numbers; Techniques for generating random numbers; Tests for Random Numbers Random-Variate Generation: Inverse transform technique; Acceptance-Rejection technique; Special properties.	8
VI	**Input Modeling :** Data Collection; Identifying the distribution with data; Parameter estimation; Goodness of Fit Tests; Fitting a non-stationary Poisson process; Selecting input models without data; Multivariate and Time-Series input models.	7
VII	**Estimation of Absolute Performance:** Types of simulations with respect to output analysis; Stochastic nature of output data; Absolute measures of performance and their estimation; Output analysis for terminating simulations; Output analysis for steady-state simulations.	8
VIII	**Verification, Calibration, and Validation; Optimization:** Model building, verification and validation; Verification of simulation models; Calibration and validation of models, Optimization via Simulation	87

Suggested Readings:

1. Jerry Banks, John S. Carson II, Barry L. Nelson, David M. Nicol: Discrete-Event System Simulation, 5th Edition, Pearson Education, 2010. (Listed topics only from Chapters1 to 12)
2. Lawrence M. Leemis, Stephen K. Park: Discrete − Event Simulation: A First Course, Pearson Education, 2006.
3. Averill M. Law: Simulation Modeling and Analysis, 4th Edition,Tata McGraw-Hill, 2007.

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:	
Further Suggestions: **None**	

Programme/Class: **M.Sc.** **(Computer science)**	Year: **Fifth**	Semester: **Ninth**
Subject: **Computer Science**		
Course Code: **B070913P**	Course Title: **Lab on System Modeling and Simulation**	
Course outcomes: 1. To know fundamental skills and knowledge required to use Mathematical tool like MATLAB or OCTAVE for the simulation and modeling 2. Describe the role of important elements of discrete event simulation and modeling paradigm. 3. Conceptualize real world situations related to systems development decisions, originating from source requirements and goals. 4. Develop skills to apply simulation software to construct and execute goal-driven system models. 5. Interpret the model and apply the results to resolve critical issues in a real world environment.		
Credits: **4**	Max. Marks: **25+75**	Min. Passing Marks:
Total No. of Lectures-Tutorials-Practical (in hours per week): **0-0-8**		
Suggested Readings: 1. Jerry Banks, John S. Carson II, Barry L. Nelson, David M. Nicol: Discrete-Event System Simulation, 5th Edition, Pearson Education, 2010. (Listed topics only from Chapters1 to 12) 2. Lawrence M. Leemis, Stephen K. Park: Discrete – Event Simulation: A First Course, Pearson Education, 2006. 3. Averill M. Law: Simulation Modeling and Analysis, 4th Edition,Tata McGraw-Hill, 2007.		
Programs: **Lab on System Modeling and Simulation**		

Edit with WPS Office

Tool Used: MATLAB or OCTAVE or any other available tool
 1. Program to illustrate Branching statements, loops, functions, additional data types, plots, arrays, inputs/outputs etc. 2. Take any function and write a code to plot with the elements of its vector representation 3. Consider a matrix A. (a) Find the determinant and inverse of A (using above mentioned tools). (b) Let B be the matrix obtained from A by rounding off to three decimal places $(1.2969\ 7 \rightarrow 1.297)$. Find the determinant and inverse of B. How do $A-1$ and $B-1$ differ? Explain how this happened. 4. Computer Generation of Random Numbers. 5. Chi-square goodness-of-fit test. 6. One-sample Kolmogorov-Smirnov test 7. Test for Standard Normal Distribution 8. Testing Random Number Generators. 9. Monte-Carlo Simulation. 10. Simulation of Single Server Queuing System. 11. Simulation of Two-Server Queuing System. 12. Simulate and control a conveyor belt system **NOTE: At least 6 Experiments out of the list must be done in the semester.**

Programme/Class: **M.Sc. (Computer science)**	Year: **Fifth**	Semester: **Tenth**
Subject: **Computer Science**		
Course Code: **B071008T**	Course Title: **Software Architectures**	
Course outcomes: 1. Argue the importance and role of software architecture in large-scale software systems. 2. Recognize major software architectural styles, design patterns, and frameworks. 3. Describe a software architecture using various documentation approaches and architectural description languages. 4. Identify and assess the quality attributes of a system at the architectural level. 5. Motivate the architectural concerns for designing and evaluating a system's architecture.		
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

		s
I	**Introduction:** The Architecture Business Cycle: Where do architectures come from? Software processes and the architecture business cycle; What makes a "good" architecture? What software architecture is and what it is not; Other points of view; Architectural patterns, reference models and reference architectures; Importance of software architecture; Architectural structures and views.	7
II	**Architectural Styles and Case Studies:** Architectural styles; Pipes and filters; Data abstraction and object-oriented organization; Event-based, implicit invocation; Layered systems; Repositories; Interpreters; Process control; Other familiar architectures; Heterogeneous architectures. **Case Studies:** Keyword in Context; Instrumentation software; Mobile robotics; Cruise control; Three vignettes in mixed style.	8
III	**Quality:** Functionality and architecture; Architecture and quality attributes; System quality attributes; Quality attribute scenarios in practice; Other system quality attributes; Business qualities; Architecture qualities. Achieving Quality: Introducing tactics; Availability tactics; Modifiability tactics; Performance tactics; Security tactics; Testability tactics; Usability tactics; Relationship of tactics to architectural patterns; Architectural patterns and styles.	7
IV	**Architectural Patterns – 1:** Introduction; From mud to structure: Layers, Pipes and Filters, Blackboard.	8
V	**Architectural Patterns – 2:** Distributed Systems: Broker; Interactive Systems: MVC, Presentation-Abstraction-Control.	8
VI	**Architectural Patterns – 3:** Adaptable Systems: Microkernel; Reflection.	7
VII	**Some Design Patterns:** Structural decomposition: Whole – Part; Organization of work: Master – Slave; Access Control: Proxy.	7
VIII	**Designing and Documenting Software Architecture:** Architecture in the life cycle; Designing the architecture; Forming the team structure; Creating a skeletal system. Uses of architectural documentation; Views; Choosing the relevant views; Documenting a view; Documentation across views.	8

Suggested Readings:

1. Len Bass, Paul Clements, Rick Kazman: Software Architecture in Practice, 2nd Edition, Pearson Education, 2003.(Chapters 1, 2, 4, 5, 7, 9)
2. Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, Michael Stal: Pattern-Oriented Software Architecture, A System of Patterns, Volume 1, John Wiley and Sons, 2007. (Chapters 2, 3.1 to 3.4)
3. Mary Shaw and David Garlan: Software Architecture- Perspectives on an Emerging Discipline, Prentice-Hall of India, 2007. (Chapters 1.1, 2, 3)
4. E. Gamma, R. Helm, R. Johnson, J. Vlissides: Design Patterns-Elements of Reusable Object-Oriented Software, Pearson Education, 1995.

Web Reference: http://www.hillside.net/patterns/

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://www.udemy.com/course/how-to-become-an-outstanding-solution-architect/?gclid=EAIaIQobChMIrvyMlMzG7gIV2AkrCh3-0gfOEAMYASAAEgLs8vD_BwE&matchtype=b&utm_campaign=LongTail_la.EN_cc.INDIA&utm_content=deal4584&utm_medium=udemyads&utm_source=adwords&utm_term=_._ag_82876580047_._ad_387389290692_._kw_%2Bsoftware+%2Barchitecture+%2Bcourse_._de_c_._dm__._pl__._ti_kwd-647104575020_._li_9040183_._pd__._   2. https://nptel.ac.in/courses/106/101/106101061/
Further Suggestions: **None**

Programme/Class: **M.Sc. (Computer science)**	Year: **Fifth**	Semester: **Tenth**
Subject: **Computer Science**		

Course Code: **B070911P**	Course Title: **Lab on Software Architectures**

Course outcomes:

1. Examine and breakdown real-world problem scenarios into structured partitions depicting static and dynamic behavior of the system using Software Architecture Requirements Capture practices.
2. Identify and formulate software requirements and behavioral models using Architectural Assessment of behavioral views by selecting appropriate architectural views.
3. Construct and justify the evolutionary system description models expressing highlevel technical architecture accommodating applicable architectural styles compatible to requirements using CASE tools.
4. Make design of different patterns using UML.

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

Total No. of Lectures-Tutorials-Practical (in hours per week): **0-0-8**

Suggested Readings:
1. Len Bass, Paul Clements, Rick Kazman: Software Architecture in Practice, 2nd Edition, Pearson Education, 2003.(Chapters 1, 2, 4, 5, 7, 9)
2. Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, Michael Stal: Pattern-Oriented Software Architecture, A System of Patterns, Volume 1, John Wiley and Sons, 2007. (Chapters 2, 3.1 to 3.4)
3. Mary Shaw and David Garlan: Software Architecture- Perspectives on an Emerging Discipline, Prentice-Hall of India, 2007. (Chapters 1.1, 2, 3)
4. E. Gamma, R. Helm, R. Johnson, J. Vlissides: Design Patterns-Elements of Reusable Object-Oriented Software, Pearson Education, 1995.
5. **Web Reference:** http://www.hillside.net/patterns/

Lab on Software Architectures

Programs:

1. To narrate Requirement Definition Document for the target system with following three areas: Problem Identification, Problem Definition, and Problem Statement
2. To narrate System Requirements Specification Document for target system with reference to the IEEE 610.12.1990 Std guidelines.
3. To narrate System Architecture Requirement Specification Document for target system with stakeholder and roles description.
4. To select appropriate Architectural View and Style and prepare Architecture Diagram for the target system.
5. To prepare Architecture Decision document describing Architectural Decisions, Software Interfaces, and behaviors along with Architectural Review.
6. To implement the target system using the Technical Architecture conforming to technology availability and scalability.
7. To create Test Plan, Test Cases and apply them to test the performance adequacy of the system implemented.

Practical's on Design Patterns

8. Use case Diagram for Librarian Scenario
9. Using UML design Abstract factory design pattern
10. sing UML design Adapter-class Design pattern
11. Using UML design Adapter-object Design pattern
12. Using UML design Strategy Design pattern
13. Using UML design Builder Design pattern
14. Using UML design Bridge Design pattern
15. Using UML design Decorator Design pattern
16. User gives a print command from a word document. Design to represent this
17. chain of responsibility Design pattern
18. Design a Flyweight Design pattern
19. Using UML design Facade Design pattern

NOTE: At least 3 Experiments from 1 to 7 and 4 experiments from 8 to 19 from the above list must be done in the semester.

Programme/Class: **M.Sc. (Computer science)**	Year: **Fifth**	Semester: **Tenth**
Subject: **Computer Science**		
Course Code: **B071009T**	Course Title: **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. H.Pham- Software Reliability Springer- Verlag, Singapore, 2000.
2. J.D. Musa et. al- Software Reliability Measurment, 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/Class: **M.Sc. (Computer science)**	Year: **Fifth**	Semester: **Tenth**
Subject: **Computer Science**		
Course Code: **B070912P**	Course Title: **Lab on Software Reliability Engineering**	
Course outcomes: 1. Develop reliable software systems. 2. Understand the fault handling and failure forecasting techniques in software systems. 3. Understand different time dependent and time independent software reliability models. 4. Design reliability models for software systems.		
Credits: **4**	Max. Marks: **25+75**	Min. Passing Marks:

82

Total No. of Lectures-Tutorials-Practical (in hours per week): **0-0-8**

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

Lab on Software Reliability Engineering

1. The student should take up the case study of Unified Library application which is mentioned in the theory, and Model it in different views i.e Use case view, logical view, component view, Deployment view, Database design, forward and Reverse Engineering, and Generation of documentation of the project.
    (a) Study of different RELIABILITY SOFTWARE TOOLS (e.g. SFRAT Software package)
    TESTING PROGRAMS
3.   Characteristics of Binomial and Poisson distributions
4.   Characteristics of Exponential and Weibull distributions
5.   Characteristics of Normal and Log-Normal distributions
6.   Determination of MTTF for series and parallel systems
7.   Evaluation of Limiting State Probabilities (LSPs)
8.   Evaluation of basic probability indices for series and parallel systems
9.   Parametric Boot-Strap estimation and finding best parameters
10. Chi-Square Goodness of Fit
11. Determination of Covariance, Correlation and Cross-Correlation coefficients
12. Neural Network design to Block box models
13. Testing of sampling methods
14. Characteristics of Histogram, Scatter diagram, Process Flow diagram and Pareto diagram

NOTE: At least 6 Experiments from the above list must be done in the semester.

Programme/Class: **M.Sc. (Computer science)**	Year: **Fifth**	Semester: **Tenth**
Subject: **Computer Science**		
Course Code: **B071010T**	Course Title: **Software Security Engineering**	

Course outcomes:
1. To demonstrate knowledge of the distinction between critical and non-critical systems.
2. To demonstrate the ability to manage a project including planning, scheduling and

risk assessment/management.
3. To demonstrate an understanding of the proper contents of a requirements for secure software.
4. To demonstrate proficiency in rapid software development techniques.

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**: System Complexity, Threats to Software Security, Sources of Software Insecurity, The Benefits of Detecting Software Security Defects Early, Managing Secure Software Development.	7
II	**What Makes Software Secure**: Introduction, Defining Properties of Secure Software, How to Influence the Security Properties of Software, How to Assert and Specify Desired Security Properties	8
III	**Requirements Engineering for Secure Software**: Introduction, Misuse and Abuse Cases, The SQUARE Process Model, SQUARE Sample Outputs	7
IV	**Requirements Elicitation and Prioritization**: Overview of Several Elicitation Methods, Elicitation Evaluation Criteria, Requirements Prioritization	8
V	**Secure Software Architecture and Design**: Introduction, Software Security Practices for Architecture and Design: Architectural Risk Analysis, Software Security Knowledge for Architecture and Design: Security Principles, Security Guidelines, and Attack Patterns	8
VI	**Considerations for Secure Coding and Testing**: Introduction, Code Analysis, Coding Practices, Software Security Testing, Security Testing Considerations Throughout the SDLC	7
VII	**Security and Complexity: System Assembly Challenges**: Introduction, Security Failures, Functional and Attacker Perspectives for Security Analysis, System Complexity Drivers and Security, Deep Technical Problem Complexity Mitigations	7
VIII	**Governance, and Managing for More Secure Software**: Introduction, Governance and Security, Adopting an Enterprise Software Security Framework, How Much Security Is Enough?, Security and Project , Maturity of Practice	8

Suggested Readings:
1. Len Bass, Paul Clements, Rick Kazman: Software Architecture in Practice, 2nd Edition, Pearson
2. Julia H. Allen, Sean J. BarnumRobert J. Ellison, Gary McGraw, Nancy R. Mead**,** Software Security Engineering: A Guide for Project Managers, Pearson Education.
3. Mark Merkow and Lakshmikanth Raghavan, Secure and Resilient Software, , CRC 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 Software Engineering

Suggested equivalent online courses:

Further Suggestions:
None

Programme/Class: **M.Sc. (Computer science)**	Year: **Fifth**	Semester: **Tenth**
Subject: **Computer Science**		
Course Code: **B070913P**	Course Title: **Lab on Software Security Engineering**	

85

Edit with WPS Office

Course outcomes:		
1. Implement the cipher techniques 2. Develop the various security algorithms 3. Use different open source tools for network security and analysis		
Credits: **4**	Max. Marks: **25+75**	Min. Passing Marks:
Total No. of Lectures-Tutorials-Practical (in hours per week): **0-0-8**		

Suggested Readings:
  1. Len Bass, Paul Clements, Rick Kazman: Software Architecture in Practice, 2nd Edition, Pearson
  2. Julia H. Allen, Sean J. BarnumRobert J. Ellison, Gary McGraw, Nancy R. Mead, Software Security Engineering: A Guide for Project Managers, Pearson Education.
  3. Mark Merkow and Lakshmikanth Raghavan, Secure and Resilient Software, , CRC Press.		

<div align="center">

Lab on Software Security Engineering

</div>

1. Implement the following SUBSTITUTION & TRANSPOSITION TECHNIQUES concepts:
    a) Caesar Cipher
    b) Playfair Cipher
    c) Hill Cipher
    d) Vigenere Cipher
    e) Rail fence – row & Column Transformation
2. Implement **any two** of the following algorithms
    a) DES
    b) RSA Algorithm
    c) Diffiee-Hellman
    d) MD5
    e) SHA-1
3. Implement the Signature Scheme - Digital Signature Standard
4. Demonstrate how to provide secure data storage, secure data transmission and for
5. creating digital signatures (GnuPG)
6. Setup a honey pot and monitor the honeypot on network (KF Sensor)
7. Installation of rootkits and study about the variety of options
8. Perform wireless audit on an access point or a router and decrypt WEP and WPA.
9. (Net Stumbler)
10. Demonstrate intrusion detection system (ids) using any tool (snort or any other s/w)

NOTE: At least 5 Experiments from the above list must be done in the semester.

Specialization Group C: Data Science

Paper Code	Paper Title	Core/Elective	Theory/ Practical	Credit	
B070914T	Foundation on Data Science	Core	Theory	4	9
B070915P	Lab on Foundation on Data Science	Core	Practical	4	9
Choose Electives I from the list					
B070904T	Machine Learning Techniques	Theory Elective	Theory	4	9
B070916T	Statistics for Data Science	Theory Elective	Theory	4	9
B070917P	Lab on Machine Learning Techniques	Lab Elective	Practical	4	9
B070918P	Lab on Statistics for Data Science	Lab Elective	Practical	4	9
Choose Electives II, III from the list					
B071014T	Data Visualization	Theory Elective	Theory	4	10
B071015T	Big Data	Theory Elective	Theory	4	10
B071016T	Natural Language Processing	Theory Elective	Theory	4	10
B071017P	Lab on Data Visualization	Lab Elective	Practical	4	10
B071018P	Lab on Big Data	Lab Elective	Practical	4	10
B071019P	Lab on Natural Language Processing	Lab Elective	Practical	4	10

Programme/Class: **Master in Science (Computer**	Year: **Fifth**	Semester: **IX**

Science)		

Subject: **Computer Science**

Course Code: **B070904T**	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.	7

	Comparing learning algorithms: cross-validation, learning curves, and statistical hypothesis testing.	
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. Logisitic 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: **Master in Science (Computer Science)**	Year: **Fifth**	Semester: **IX**
Subject: **Computer Science**		
Course Code: **B070916T**	Course Title **Statistics for Data Science**	

Course Outcomes: After the completion of the course, the students will be able:
CO1: To learn advanced statistical technique and apply them to the analysis of real data sets.
CO2: To analyse data and draw inferences from data analysis.
CO3: To learn and apply data mining techniques on data sets.
CO4: To make estimation and prediction with the given data.

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 to Statistics: fundamental statistical concepts, examining distributions, describing categorical data, constructing confidence intervals, simple tests of hypothesis, Chi-Square as a test of independent, Chi-square as a Test of goodness of fit; Analysis of Variance (ANOVA) : one-way ANOVA, multiple comparisons, two-way ANOVA with and without interactions; Multivariate Analysis of Variance (MANOVA): Type 1 errors, relation to ANOVA, one-way MANOVA, two-way MANOVA, Hypothesis testing techniques; Factor analysis, cluster analysis.	8
II	Regression Models: Least squares and linear regression, Ordinary least squares; Regression to the mean; Linear regression; Residuals; Regression inference; Multivariable regression: Multivariate regression; Residual variation and diagnostics; Multiple variables , Non-linear Transformations of the Predictors, Qualitative Predictors.	8

III	Multiple Regression Analysis; Dummy Variable Regression Models; Multi-collinearity, Heteroscedasticity, Autocorrelation; Econometric Modelling: Model Specification and Diagnostic Testing; Correlation and Covariance Analysis; Canonical Analysis, Canonical Roots/variates;	8
IV	Extension of regression analysis: Ridge Regression, The Lasso Nonlinear Regression Models: Approaches to Estimating Nonlinear Regression models.	8
V	Generalized linear models: Logistic Regression, Binary outcomes, Count outcomes, Multiple Logistic Regression.	7
VI	Classification: Using Bayes' Theorem for Classification, Procedure of Discriminant Analysis, Linear Discriminant Analysis, Estimating Misclassification Probabilities, Quadratic Discriminant Analysis; Cluster Analysis: Measures of Association for Continuous Variables, Measures of Association for Binary Variables, Agglomerative.	7
VII	Hierarchical Clustering, Ward's Method, K-Means Procedure, K-Nearest-Neighbours; Maximum Likelihood Estimation Method;	7
VIII	Resampling Methods: sample goal, sample size, selection criteria, standard error, Cross Validation, **Leave-one-out-cross-validation, k-fold cross-validation,**Bootstrapping, Challenges, Jackknife Resampling, Permutation Testing.	7

 Suggested Readings:
1. D. Gujarati, "Basic Econometrics," McGraw Hill, 2011.
2. T. Hastie and T. Robert, "An Introduction to Statistical learning with application in R," Springer: New York, 2014.
3. T. Hastie, T. Robert,and J. **Friedman**, "The Elements of Statistical Learning Data Mining, Inference, and Prediction," Second Edition, Springer: New York, 2009.
4. Richard I. Levin and David S. Rubin, "Statistics for Management," Seventh Edition, Pearson, 1998.

 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 class12[th].

Suggested equivalent online courses:

Further Suggestions: Programs:

Programme: **Master in Science (Computer Science)**	Year: **Fifth**	Semester: **IX**
Subject: **Computer Science**
Course Code: **B070917P**		Course Title: **Lab on 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**	Max. Marks: **25+75**	Min. Passing Marks:

Total No. of Lectures-Tutorials-Practical (in hours per week): **0-0-8**

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

Lab on on Machine Learning Techniques

List of Practical in AI with python:

1. Write a program to implement k-Nearest Neighbour algorithm to classify the iris data set. Print both correct and wrong predictions. Python ML library classes can be used for this problem.
2. Write a program to construct a Bayesian network considering medical data. Use this model to demonstrate the diagnosis of heart patients using standard Heart Disease Data Set. You can use Python ML library classes/API.

3. Assuming a set of documents that need to be classified, use the naïve Bayesian Classifier model to perform this task. Calculate the accuracy, precision, and recall for your data set.
4. Implement and demonstrate the FIND-S algorithm for finding the most specific hypothesis based on a given set of training data samples.
5. Write a program to demonstrate the working of the decision tree based ID3 algorithm. Use an appropriate data set for building the decision tree and apply this knowledge to classify a new sample.
6. Implement the non-parametric Locally Weighted Regression algorithm in order to fit data points. Select appropriate data set for your experiment and draw graphs.
7. Implement the Viterbi Algorithm in Hidden Markov Model.
8. Implements the Forward-backward EM algorithm for training the parameters of HMM's
9. Implement the HMM for pattern classification of English vowels.

Programme/Class: **M.Sc. (Computer science)**	Year: **Fifth**	Semester: **IX**
Subject: **Computer Science**		
Course Code: **B070918P**	Course Title: **Lab on Statistics for Data Science**	
Course Outcomes: After the completion of the course, the students will be able: CO1: To learn advanced statistical technique and apply them to the analysis of real data sets. CO2: To analyse data and draw inferences from data analysis. CO3: To learn and apply data mining techniques on data sets. CO4: To make estimation and prediction with the given data.		
Credits: **4**	Max. Marks: **25+75**	Min. Passing Marks:
Total No. of Lectures-Tutorials-Practical (in hours per week): **0-0-8**		
Suggested Readings: 2. Jay Liebowitz, —Big Data And Business Analytics Laboratory, CRC Press.		
Q1. A dataset that contains total 1000 votes from different races (Asian, Black, Hispanic, White, and other) and parties (Democrat, Independent, and Republican) is given below. Write Python/ R Program for a) Create Observed table and Expected table b) Calculate the Chi-Square value and Critical value		

Index	democrat	independent	republican	row_totals
asian	21	7	32	60
black	65	25	64	154
hispanic	107	50	94	251
other	15	8	15	38
white	189	96	212	497
col_totals	397	186	417	1000

Q2. Write a Python / R program to perform following with suitable dataset.

a)	One-way ANOVA

b)	Two-way ANOVA

Q3. Write a Python / R program to perform following with suitable dataset.

a)	Linear Regression with one variable

b)	Multiple Linear Regressions.

Q4. Write a Python / R program for Ridge regression.

Q5. Write a Python / R program for Lasso regression.

Q6. Write a Python / R program to perform following with suitable dataset.

a)	Logistic Regression with one variable.

b)	Logistic Regression with multiple variables.

Q7. Write a Python / R program to perform binary classification using Naïve Bayes algorithm by using a suitable dataset.

Q8. Write a Python / R program to perform binary classification using K-Nearest-Neighbors algorithm by using a suitable dataset.

Q9. Write a Python / R program to predict some real value using K-Nearest- Neighbors algorithm by using a suitable dataset.

Q10. Write a Python / R program to implement K- Means clustering algorithm by using a suitable dataset.

Programme/Class: **M.Sc. (Computer science)**	Year: **Fifth**	Semester: **Tenth**
Subject: **Computer Science**		
Course Code: **B071014T**	Course Title: **Data Visualization**	

Course outcomes:
By the completion of this course, learners will be able to:
CO1: Know the basics of data visualization
 CO 2: Design and create data visualizations.
CO 2: Conduct exploratory data analysis using visualization.
CO 3: Craft visual presentations of data for effective communication.
CO 4: Apply data transformations such as aggregation and filtering for visualization.
CO 6: Identify opportunities for application of data visualization in various domains.

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** to data and its visualization, Importance of analytics, Visual Representations and Interaction Technologies, Develop a new suite of visual paradigms that support the analytical reasoning process, Visual representation principles, types of data, address scale and information complexity, knowledge discovery through information synthesis, and facilitate analytical reasoning.	7
II	**Problem solving with visual analytics:** visual analytics pocess, building blocks of visual analytics, challenges and opportunities, spatio-tempoal (space and time) visual analytics, dependencies between observations, Uncertainties.	8
III	**Exploratory data analysis (EDA)**, Types of EDA, Data formats, Univariate non-graphical EDA, Central tendency, Spread, Skewness and kurtosis, Univariate graphical EDA, Histograms, Stem-and-leaf plots, Quantile-normal plots.	7
IV	**Multivariate non-graphical EDA**, Cross-tabulation, Correlation and covariance, Covariance and correlation matrices, Multivariate graphical EDA.	8
V	**EDA using R**, EDA vs. Classical data analysis, EDA goals, Role of graphics, Data treatment, Scatter plots, Histograms, Probability plots, Residual Plots, Box plots, Block plots, Interpretation of plots.	8
VI	**Data visualization in Python**: Matplotlib API, Plot types, legends and annotations, Plotting functions with Pandas, Bokeh, MayaVi.	7
VII	**Gephi**: fundamentals, acquiring data, importing data into Gephi, organize data with layouts, filtering, size, color, colormaps, color channels, facets and views, Juxtapose and Coordinate Views, Partition into Views, Superimpose Layers	7
VIII	**Tableau**: introduction to Tableau, data preperation, sample dataset, workspace, setting up data connectors, data tables, data types, join, union, chart types, table calculations, maps.	8

Suggested Readings:
1. Daniel A. Keim, Florian Mansmann, Andreas Stoffel, Hartmut Ziegler, "Visual Analytics", University of Konstanz, Germany, 2014.
2. Andy Kirk, Data Visualization A Handbook for Data Driven Design, Sage Publications, 2016
3. Philipp K. Janert, Gnuplot in Action, Understanding Data with Graphs, Manning Publications, 2010.

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**

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

6. 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)

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

8. 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

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

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

		s
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, Crop 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 Directtories, 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, Micheline 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

Edit with WPS Office

Programme/Class: **M.Sc. (Computer science)**	Year: **Fifth**	Semester: **Tenth**
Subject: **Computer Science**		
Course Code: **B071016T**	Course Title: **Natural Language Processing**	

Course outcomes:
CO1: Understand the mathematical and linguistic foundations for natural language processing.
CO2: Understand approaches to syntax and semantics in NLP.
CO34: Understand approaches to discourse, generation, dialogue and summarization within NLP.
CO 4:Understand current methods for statistical approaches to machine translation.
CO 5: Understand machine learning techniques used in NLP.

Credits: **4**	**Specialization Group B: Elective**

101

	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	**Natural language and Formal language**, NLP tasks in syntax, semantics, and pragmatics, Applications such as information extraction,The problem of ambiguity, The role of machine learning in NLP, ArgMax Computation.	7
II	**WSD**: WordNet, Wordnet; Application in Query Expansion, Wiktionary; semantic relatedness, WordNet Similarity, N-grams and language models, Corpora, Unigram, Bigram, and Trigram models, Usage of N-grams, N-grams Training & Testing.	8
III	**N-grams** (cont.), Counting, Probability, Perplexity, Entropy, Smoothing techniques, Backoff methods, Class-based models. Part of Speech (POS) Tagging, POS Tagger, Chunking.	7
IV	**Parsing Algorithms**, Evidence for Deeper Structure; Top Down Parsing Algorithms, Noun Structure, Non-noun Structure, Probabilistic parsing; sequence labeling, PCFG, Probabilistic parsing: Training issues, Probabilistic parsing; inside-outside probabilities.	8
V	**Text Clustering**, Distributional Semantics, Morphology, Graphical Models for Sequence Labelling in NLP, Phonetics.	8
VI	Consonants, Vowels, Phonology, HMM and Viterbi, Forward Backward probability, Opinions on the Web, Machine Translation, Text Entailment.	7
VII	**Sentiment Analysis**, Semantic Analysis, Text coherence and discourse structure, Information extraction, Information retreival, Pronounse, Reference resolution.	7
VIII	**Precision**, Recall, F-score, Map, Semantic Relations; UNL; Towards Dependency Parsing, Universal Networking Language, Semantic Role Extraction, Baum Welch Algorithm; HMM training.	8

Suggested Readings:

1. Allen, James, Natural Language Understanding, Second Edition, Benjamin/Cumming, 1995.
2. Charniack, Eugene, Statistical Language Learning, MIT Press, 1993.
3. Jurafsky, Dan and Martin, James, Speech and Language Processing, Second Edition, Prentice Hall, 2008.
4. Manning, Christopher and Heinrich, Schutze, Foundations of Statistical Natural Language Processing, MIT Press, 1999.

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** 13. Assessment Type: Class Tests (Max. Marks 14) 14. 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) 15. Assessment Type: Assignments (Max Marks: 4) 16. 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**

Programme/Class: **M.Sc. (Computer science)**	Year: **Fifth**	Semester: **Tenth**
Subject: **Computer Science**		
Course Code: **B071017P**	Course Title: **Lab on Data Visualization**	
Course outcomes: CO1: Know the basics of data visualization CO 2: Design and create data visualizations. CO 2: Conduct exploratory data analysis using visualization. CO 3: Craft visual presentations of data for effective communication. CO 4: Apply data transformations such as aggregation and filtering for visualization. CO 6: Identify opportunities for application of data visualization in various domains.		

Credits: **4**	Max. Marks: **25+75**	Min. Passing Marks:
Total No. of Lectures-Tutorials-Practical (in hours per week): **0-0-8**		

Suggested Readings:
1. Daniel A. Keim, Florian Mansmann, Andreas Stoffel, Hartmut Ziegler, "Visual Analytics", University of Konstanz, Germany, 2014.
2. Andy Kirk, Data Visualization A Handbook for Data Driven Design, Sage Publications, 2016
3. Philipp K. Janert, Gnuplot in Action, Understanding Data with Graphs, Manning Publications, 2010.

Lab on Data Visualization

This practical course uses data from the UC Irvine Machine Learning Repository, a popular repository for machine learning datasets. In particular, we will be using the "Auto MPG Data Set" available from https://archive.ics.uci.edu/ml/datasets/Auto+MPG. **Do the following using R/ Python:**

1. How many cars and how many attributes are in the data set.

2. How many distinct car companies are represented in the data set? What is the name of the car with the best MPG? What car company produced the most 8-cylinder cars? What are the names of 3-cylinder cars? Do some internet search that can tell you about the history and popularity of those 3-cylinder cars.

3. What are the range, mean, and standard deviation of each attribute? Pay attention to potential missing values.

4. Plot histograms for each attribute. Pay attention to the appropriate choice of number of bins. Write 2-3 sentences summarizing some interesting aspects of the data by looking at the histograms.

5. Plot a scatter plot of weight vs. MPG attributes. What do you conclude about the relationship between the attributes? What is the correlation coefficient between the 2 attributes?

6. Plot a scatter plot of year vs. cylinders attributes. Add a small random noise to the values to make the scatter plot look nicer. What can you conclude? Do some internet search about the history of car industry during 70's that might explain the results.(Hint: data.mpg + np.random.random(len(data.mpg)) will add small random noise)

7. Show 2 more scatter plots that are interesting do you. Discuss what you see.

8. Plot a time series for all the companies that show how many new cars they introduces during each year. Do you see some interesting trends? (Hint: data.car name.str.split()[0] returns a vector of the first word of car name column.)

9. Calculate the pair wise correlation, and draw the heatmap with Matplotlib. Do you see some interesting correlation? (Hint: data.iloc[:,0:8].corr(), plt.pcolor() draws the heatmap.)

10. Calculate the pair wise covariance.

11. Calculate Spread, Skewness and Kurtosis of 'Weight' attribute.

NOTE: At least 8 Experiments from the above list must be done in the semester.

Programme/Class: **M.Sc. (Computer science)**	Year: **Fifth**	Semester: **Tenth**
Subject: **Computer Science**		
Course Code: **B071018P**	Course Title: **Lab on Big Data**	

Course outcomes:
1. Optimize business decisions and create competitive advantage with Big data analytics
2. Practice java concepts required for developing map reduce programs.
3. Impart the architectural concepts of Hadoop and introducing map reduce paradigm.
4. Practice programming tools PIG and HIVE in Hadoop eco system.
5. Implement best practices for Hadoop development.

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

Total No. of Lectures-Tutorials-Practical (in hours per week): **0-0-8**

Suggested Readings:
1. Jay Liebowitz, ─Big Data And Business Analytics Laboratory, CRC Press.
4.

Lab on Big Data

1. Installation of VMWare to setup the Hadoop environment and its ecosystems
2. a. Perform setting up and Installing Hadoop in its three operating modes.
   (ii.)   Standalone.
   (iii.)   Pseudo distributed.
   (iv.)   Fully distributed.
   b. Use web based tools to monitor your Hadoop setup.
3. Implementing the basic commands of LINUX Operating System – File/Directory creation, deletion, update operations.
4. Implement the following file management tasks in Hadoop:

   (i.)   Adding files and directories
   (ii.)   Retrieving files
   (iii.)   Deleting files
   Hint: A typical Hadoop workflow creates data files (such as log files) elsewhere and copies them into HDFS using one of the above command line utilities.
5. Run a basic word count Map Reduce program to understand Map Reduce Paradigm.
6. Write a Map Reduce program that mines weather data. Hint: Weather sensors collecting data every hour at many locations across the globe gather a large volume of log data, which is a good candidate for analysis with Map Reduce, since it is semi structured and record-oriented
7. Implement matrix multiplication with Hadoop Map Reduce.
8. Installation of PIG.
9. Write Pig Latin scripts sort, group, join, project, and filter your data.
10. a. Run the Pig Latin Scripts to find Word Count
    b. Run the Pig Latin Scripts to find a max temp for each and every year.
11. Installation of HIVE.
12. Use Hive to create, alter, and drop databases, tables, views, functions, and indexes.

   NOTE: At least 8 Experiments from the above list must be done in the semester.

Programme/Class: **M.Sc. (Computer science)**	Year: **Fifth**	Semester: **Tenth**
Subject: **Computer Science**		
Course Code: **B071019P**	Course Title: **Lab on Natural Language Processing**	

 Course outcomes:
CO1: Understand the mathematical and linguistic foundations for natural language processing.

CO2: Understand approaches to syntax and semantics in NLP.
CO34: Understand approaches to discourse, generation, dialogue and summarization within NLP.
CO 4:Understand current methods for statistical approaches to machine translation.
CO 5: Understand machine learning techniques used in NLP.

Credits: 4	Max. Marks: 25+75	Min. Passing Marks:
Total No. of Lectures-Tutorials-Practical (in hours per week): 0-0-8		

Suggested Readings:
 1. Jay Liebowitz, —Big Data And Business Analytics Laboratory, CRC Press.

Virtual Lab on Natural Language Processing
https://nlp-iiith.vlabs.ac.in/

List of experiments aviable at virtual Lab are given below:

11. Word Analysis

12. Word Generation

13. Morphology

14. N-Grams

15. N-Grams Smoothing

16. POS Tagging: Hidden Markov Model

17. POS Tagging: Viterbi Decoding

18. Building POS Tagger

19. Chunking

20. Building Chunker

NOTE: At least 8 Experiments from the above list must be done in the semester.

Guidelines for Preparing Literature Survey

The guidelines taken from [1] are simply for illustration purpose to write the literature survey. You can also talk to your supervisor for any guidance during writing the literature survey..

A literature survey is a kind of review of academic sources on a particular topic. It provides a general idea of existing knowledge, allowing you to classify related theories, methods, and any gaps in the existing research.

The literature survey is normally conducted to put your research within existing knowledge while writing dissertation, thesis or research paper. The literature survey provides you a chance to:

- Show your acquaintance with the topic
- Build a theoretical framework and methodology for the research
- Align yourself with respect to other researchers
- Give an idea about how your research addresses a gap

It is also possible that you are writing a literature survey as a stand-alone researcher. In this case, the intention is to judge and demonstrate your knowledge about the current state of research to make academic debates on a specific topic.

In both the above cases, the process of conducting a literature survey remains the same. To write a literature review, it involves finding pertinent publications (such as books, conference papers and journal articles), decisively analyzing them, and explaining what you found. This process involves major five steps:

1. **Search** for relevant literature
2. **Evaluate** and select sources
3. **Identify** themes, debates and gaps
4. **Outline** the structure
5. **Write** your literature survey

A literature survey not only summarizes the existing knowledge on a specific topic, it also analyzes, synthesizes, and critically evaluates the existing sources to give a clear picture of the state of knowledge on the subject.

Step 1: Search for relevant literature
(a) Definite topic is required before starting the literature survey.
(b) Make a list of similar or related keywords linked to your research question. Try to add new keywords in this list if you find any new keywords in the process of your literature search.

(c) Search for relevant sources: The keywords obtained from step (b) above are used for searching the sources. Some of the useful databases to search for journals and articles include:
- Library catalogue of your University or college
- Inspec (physics, engineering and computer science)
- DBLP (basic sciences, engineering and computer science)
- Project Muse (humanities and social sciences)
- Medline (life sciences and biomedicine)
- Econlite (economics)
- Google Scholar
- JSTOR
- EBSCO

You can use boolean operators like AND, OR and NOT to refine your searches.
You can also read the abstract and identify that whether it is useful for your specified topic. If it is useful than check its bibliographic references to back track and identify previous relevant sources.

Step 2: Evaluate and select sources

Before selecting the correct source of knowledge you should evaluate which article or source is useful for your definite topic of research. While reading any article, ask the following question to yourself:

- What kind of problem or question is addressed by the author?
- What are the key features and concepts of the article?
- Is the article addresses some new theories, models and methods or uses some established frameworks?
- What are the results and conclusions of the article?
- How the article helps you to relate to other literature in the field?
- Is article helps in your topic? What are its key theories, concepts and arguments?
- What are the strengths and weaknesses of the research?

While selecting any article, must sure that you read any landmark studies and major theories in your field of research so that its contents are trustworthy. This can be checked by the citation count of the article using Google Scholar. A high citation count specifies that article is significant in the field of research and shall be included in the literature survey.

Step 3: Identify themes, debates, and gaps

Before starting the argument and structure of literature survey, you should be aware of the connections and relationships between the sources you have read so far. Based on your reading and notes, you can come across for:

- **Trends and patterns (in theory, method or results):** Is approaches you read are

still useful or becomes less popular over time?

- **Themes:** Which kind of questions or concepts reappear across the literature?
- **Debates, conflicts and contradictions:** Is different sources disagree to some extent?
- **Essential publications:** Is there any theories or studies that are prominent and changed the direction of the field?
- **Gaps:** Still is there any possibility to explore new things from the read literature you have read and still missing in the literature? Is there any weaknesses that need to be addressed?

The above steps will let you to prepare the structure of your literature survey and (if applicable) give you an idea about how your own research will contribute to existing knowledge.

Step 4: Outline your literature survey's structure

Prepare a rough idea of your approach before writing the literature survey. There are many ways to organize the body of a survey. Depending on the length of your literature survey, many of these strategies are clubbed.

(a) Prepare a list of development of sources or topics and arrange them with respect to time. Critically analyze, show key points and summarize them in a brief
(b) If you found some recurring themes, put them in subsection of central theme and concentrate to different aspects of the topic.
(c) If you read the sources from different fields or discipline, than compare the results and conclusions arises from the different methods.
(d) Any literature survey is a backbone of new theoretical framework. While writing the literature survey, you should converse different theories, models and key points of read topics.
(e) At last and not the least, argue yourself for the significance of certain theoretical methods, or may combine different concepts to create your own framework for your future research.

Step 5: Write your literature survey
The literature survey should have an introduction, main body of the text followed by conclusion. The contents on each can be your choice depending upon the requirement of your research.

Reference:
1. https://www.scribbr.com/dissertation/literature-review/ (Author: Shona McCombes)

Common Guidelines for Course Code: B070806R and Course Code: B071020R

Research Project and Dissertation Guidelines for VIII and X Semester

1. **Objectives**
- To facilitate the student to independently formulate and solve a social, philosophical, commercial, or technological problem and present the results in written and oral form.
- To render students to the real life problems.
- To provide opportunities to students to interact with people and present them confidently.

2. **Types of Project**
   The students are expected to work on Research Oriented Project.

   However, it is not mandatory for a student to work on a real-life project. The student can formulate a project problem with the help of his Guide and submit the project proposal of the same. **Approval of the project proposal is mandatory.** If approved, the student can commence working on it, and complete it. Use the latest versions of the software packages or tools for the development of the project.

3. **Software and Broad Ideas of Application but not limited to:**

- **Languages** - C, C++, Java, VC++, C#, R,Python
- **Scripting Languages** - PHP,JSP, SHELL Scripts (Unix), TcL/TK
- **.NET Platform** - F#,C#. Net, Visual C#. Net, ASP.Net
- **Middle Ware(Component) Technologies** - COM/DCOM, Active-X, EJB
- **Front-End/GUI Tools** - .Net Technologies, Java

Edit with WPS Office

- **Back-End/DBMS** - Oracle, SQL Plus, MY SQL, SQL Server
- **UNIX Internals** - Device Drivers, RPC, Threads, Socket programming
- **Real time Operating Systems/Embedded Skills** - LINUX, Raspberry Pi, Arduino.
- **Application and Research Areas** - Financial / Insurance / Manufacturing / Multimedia / Computer Graphics / Instructional Design/ Database Management System/ Internet / Intranet / Computer Networking- Communication Software development/ E-Commerce/ ERP/ MRP/ TCP-IP programming / Routing protocols programming/ Socket programming / Algorithm design for new problems, soft computing, machine learning, data science, deep learning, data visualization, natural language processing, software architecture etc.

  - Programming Language/ Application Software/Operating System etc could be any latest technology prevailing / being used as per the will of the student with the consent of the guide.

4. **Eligibility of the Guide**

   Guide should be a regular teacher of the University/College/Higher Education Institute. Student can also do the project under the guidance of regular teacher of Institute of National Importance with the consent of the enrolled college/institute where the student is studying.

5. **Introduction to the Project**
   The student should include the details in the project diary, in which they will record the progress of their project throughout the course. The project report should be documented with scientific approach to the solution of the problem that the students have sought to address. The project report should be prepared in order to solve the problem in a methodical and professional manner, making due references to appropriate techniques, technologies and professional standards. The project report should contain enough details to enable examiners to evaluate the work. The important points should be highlighted in the body of the report, with details often referred to appendices.

6. **Structure and Format of the Project and Dissertation**
   Research Project submitted in Semester VIII shall be in spiral binding. The dissertation submitted in Semester X shall be hard bound with golden embossing. Students will be evaluated based on the Research project in VIII and Dissertation in X semester independently.

(i)   **Title Page:**
       Sample format of Title page is given below. Students should follow the given format.

(All the text should be in Times New Roman)

<TITLE OF THE PROJECT OR DISSERTATION>
(NOT EXCEEDING 2 LINES, 24 BOLD, ALL CAPS)

A <Project Report OR Dissertation> (12 Bold)

(ii) **Original Copy of the Approval Proforma of the Project Proposal:**
Sample Proforma of Project Proposal is given below. Students should follow the given format.

PROFORMA FOR THE APPROVAL OF PROJECT/DISSERTATION PROPOSAL

(Note: All entries of the proforma of approval should be filled up with appropriate and complete information. Incomplete proforma of approval in any respect will be rejected)

Roll no:…………………..

1. Name of the Student

……………………………………………………….

2. Title of the Project

……………………………………………………….

3. Name of the Guide

…………………………………………

4. Teaching experience of the Guide

………………………………

Signature of the Student   Signature of the Guide

Date……………………   Date:………………

Signature of the Project Coordinator

Date………….

(iii) **Certificate of Authenticated work:**
Sample format of Certificate of Authenticated work is given below.
Students should follow the given format.

UNIVERSITY/COLLEGE NAME (14 BOLD, CAPS)
(Affiliated to University Name) (13, bold, italic)
CITY NAME-PINCODE (13 bold, CAPS)

DEPARTMENT NAME (14 BOLD, CAPS)

College Logo
CERTIFICATE (14 BOLD, CAPS, underlined, centered)

This is to certify that the project entitled, "**Title ofThe Project/Dissertation**", is bonaficd work of **NAME OF THE STUDENT** bearing **Roll No.** submitted in partial fulfillment of the requirements for the award of degree of BACHELOR OF SCIENCE IN RESEARCH OR MASTER OF SCIENCE IN COMPUTER SCIENCE from **University Name**. (12, times new roman, justified)

Name of Internal Guide (12 bold)   Name of Coordinator

(Don't write names of lecturers or HOD)

External Examiner

Date: College Seal

(iv) **Certificate from other Institute of National Importance** (to be issued by the HEI and the photocopy of the certificate is to be attach)

(v) **Abstract**
This should be one/two short paragraphs (100-150 words total), summarizing the project/dissertation work. It will not be a re-statement of the original project outline. A suggested flow is background, project aims and main achievements. From the abstract, a reader should be able to determine if the project is of interest to them and, it should present results of which they may wish to know more details.

(Project/dissertation Abstract page format)

Abstract (20bold, caps, centered)

Content goes here (12, justified)

Note: Entire document should be with 1.5 line spacing and all paragraphs should start with 1 tab space.

(vi) **Acknowledgements**

114

This should express student's gratitude to those who have helped in the preparation of project.

ACKNOWLEDGEMENT (20, BOLD, ALL CAPS, CENTERED)

The acknowledgement should be in times new roman, 12 font with 1.5 line spacing, Justified.

(vii)   Declaration

(Declaration page format)

DECLARATION (20 bold, centered, allcaps)
Content (12, justified)

I here by declare that the project entitled, **"Title of the Project/dissertation'** done at **[name of place where projects is done]** has not been in any case duplicated to submit to any other university for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other universiny.

The project is done in partial fulfilment of the requirements for the award of degree of **BACHELOR OF SCIENCE IN RESEARCH OR MASTER OF SCIENCE IN COMPUTER SCIENCE** to be submitted as **[VIII OR X]** semester project as part of our curriculum.

(viii)   Table of Contents

The table of contents gives the readers a view of the detailed structure of the report. The students would need to provide section and subsection headings with associated pages. The formatting details of these sections and subsections are given below.

TABLE OF CONTENTS (20bold, caps, centered)

Should be generated automatically using word processing software.

(ix)   List of Tables

List of all the tables in the project along with their page numbers.

> List of Tables (20 bold, centered, Title Case)
>
> Should be generated automatically using word processing software.

(x)   List of Figures

List of all the figures, graphs, charts etc. in the project along with their page numbers.

> List of Figures (20 bold, centered, Title Case)
>
> Should be generated automatically using word processing software.

Chapter 1: Introduction

The introduction has several parts as given below:

- Background: A brief detail of background and framework of project and its relation to work done in the area.
- Objectives: Point wise statement of the aims and objectives of the project
- Purpose, Scope and Applicability: The description of Purpose, Scope, and Applicability are given below:
  - o Purpose: Describe the topic of the project on the basis of why this project is being done. How this project improve the existing system.
  - o Scope: Describe methodology, assumptions and limitations.

- o   Applicability: State the application of project.
- Achievements: Explain what kind of purpose is achieved after completion of project.
- Organization of Report: Summarize remaining chapters of the project report.

> (Project Introduction page format)
>
> **Chapter 1**
> <div align="center">

Introduction (20 Bold, centered)
</div>
>
> Content or text (12, justified)
>
> Note: Introduction has to cover brief description of the project with minimum 4 pages.

Chapter 2: Literature Review OR Survey of Technologies

In this chapter survey of technologies for application oriented project should demonstrate the student awareness and understanding of available technologies OR literature survey is required for research oriented project. The student should give the detail of all the related literature/technologies that are necessary to complete the project. The student should present a comparative study of all those technologies/literature.

Chapter 3: Requirements and Analysis (For Application Oriented) OR [Title of Research Working Chapter]

Chapter 4: System Design (For Application Oriented) OR [Chapter related to Research Work]

Chapter 5: Implementation and Testing

- **Implementation Approaches:** Define the plan of implementation, and the standards or standard data sets used in the implementation.
- **Coding Details and Code Efficiency:** Students not need include full source code, instead, include only the important codes (design of new data structure, algorithms, applets code, forms code etc). The program code should contain comments needed for explaining the work a piece of code does. Comments may be needed to explain why it does it, or, why it does a particular way. The student can explain the function of the code with a shot of the output screen of that program code. The student should explain how the code is efficient and how the students have handled code optimization.
- **Testing Approach**
- **Modifications and Improvements**

Chapter 6: Results and Discussion

- **Test Reports:** Student should provide the test results and reports based on

the test cases to show that it works fine in different conditions of input.

- **User Documentation:** In this section, working of the software should be explained; also explain its different functions with screen shots. The user document should be like a manual.

Chapter 7: Conclusions and Future Work
The conclusions shall be summarized with in 2 or 3 pages. This chapter mainly focuses on:

- Limitations of the Proposed System OR Research
- Future Scope describes new areas of investigation and parts of the current work that was not completed due to time constraints and/or problems encountered.

(xi)   References

In this, students acknowledge the work of others that they have used or adapted in their own work. Student can follow the given standard for the references for books, journals, and online material. The citation is mandatory in both the reports.

Eg.
Lipson, Charles (2011). Cite right : A quick guide to citation styles; MLA, APA, Chicago, the sciences, professions, and more (2nd ed.). Chicago [u.a.]: University of Chicago Press. p. 187. ISBN 9780226484648.

(xii)   Glossary

If any acronyms, abbreviations, symbols, or uncommon terms is used in the project report then their meaning should be explained where they first occur.

(xiii)   Appendices
Appendix include some further details like results, mathematical derivations, certain illustrative parts of the program code (e.g., class interfaces), user documentation etc.

7. Evaluation
- During the project/dissertation work, its progress will be monitored, on fortnightly/monthly basis, by the internal guide.
- 2-3 copies of Project/Dissertation Report to be submitted to department (1 copy to be retained by department, 1 copy for student, 1 copy for evlaluation)
- End Examination shall be based on Project Report/Dissertation work, Presentation, Viva, and Demonstration of the software.
- Project/Dissertation carries 6 Credit Points.

Duration:

Evaluation in VIII and X semester separately		
Type of evaluation	Total time	Max. Marks

Edit with WPS Office

Presentation	10 minutes	25
Viva	10 minutes	20
Demonstration	5 minutes	20
Report/Dissertation checking	5 minutes	35
Total Time/Max. Marks	**30 minutes**	**100**

Format of Certificate of Evaluation
Certificate of Evaluation (14 point, Times, Bold)

This is to certify that the undersigned have assessed and evaluated the project work titled "………………………." submitted by the following student(s).
1.
2.
3.

The project report has been accepted/ rejected for the partial fulfillment of B.Sc. in Research OR M.Sc. (Computer Science) programme.

Signature of the examiner
Name of the examiner

8. Project/Dissertation Viva Voice

Student may be asked to write code for some segment of the problem during VIVA to check his coding capabilities.