HSNC University Mumbai

(2021-2022)

Ordinances and Regulations

With Respect to

Choice Based Credit System

(CBCS)

For the Programmes Under

The Faculty of Science and Technology

For the Course

Computer Science

Curriculum – Second Year Postgraduate Programme

Semester - III and Semester - IV

2021-2022

Section D

Computer Science

Part 1- Preamble

This syllabus is an extension of the syllabus for semester - I and semester – II of MSc Computer Science of HSNC University, Mumbai, which came into existence in the academic year 2020-2021. This two-year M. Sc. programme is designed by experts from Academia, Industry and Alumni to develop skilled Computer Scientist, who can progress to diverse fields of Computer Science interests that include industry, research, teaching, and entrepreneurship.

MSc Computer Science syllabus is an attempt to include following ideas, among other things, into practice.

1. Course Objective

- Bring a new approach to syllabus, not a revision of the existing syllabus.
- Create a unique identity for MSc in Comp Science distinct from similar degrees in other related subjects.
- Recommend provision for specialization in MSc Computer Science degree.
- Offers focus on core Computer Science subjects.
- Incorporate advanced and most recent trends.
- Identify and nurture research temper among students.
- Offer provision for internship with industry.
- Focus, as far as possible, only on open-source software.

This syllabus for Semester I and Semester II has tried to initiate steps to meet these goals. By extending the syllabus to Semester III and Semester IV, it is assumed that these goals will be met to a larger extent.

2. Process adopted for curriculum designing

The final programme was outlined after frequent discussions, meetings, brainstorming sessions and electronic interactions with academic, alumni and industry partners.

3. Salient features Syllabus made more relevant

This syllabus for the semester - III and semester – IV has tried to continue the steps initiated in the semester- I and semester –II to meet the goals set. This proposes two core compulsory

subjects in semester III. The student has to continue with the tracks they have taken in the semester II as elective subjects. Semester – IV will have one subject as specialization out of the two electives he or she has been pursuing since the semester – II. That means, there will be four specializations in the semester IV as mentioned below:

- Artificial Intelligence and Machine Learning
- Cloud Computing
- Big Data Engineering
- Human Computer Interaction

The syllabus also offers an internship and project implementation in the semester – IV, each of which has weights equivalent to a full course. By introducing different electives as tracks in semester –II, espousing more of that tracks in the semester –III and offering the opportunity to choose the specialization based on the tracks pursed in semester –IV will give the student the added advantage of high level competency in the advanced and emerging areas of computer science. This will definitely equip the student with industry readiness as internship in an IT or IT-related organization gives a practical exposure to what is learned and what is practiced. The strong foundation given in the core courses in different semesters will give enough confidence to the learner to face and adapt to the changing trends and requirements of industry and academia.

As one can easily notice, the syllabus offers lots of emphasis on student driven learning and learning through experience. Research is embedded in the course structure. By introducing Research Methodology in semester – I, Case study (Self-Learning Evaluation, Project Assignment), Project Proposal in semester – III, Internship and Project Dissertation in semester – IV (which together has a weightage equivalent to almost two theory courses), the syllabus prepares a strong army of budding computer science researchers. The syllabus designed on the firm believe that by focusing on student driven research on cutting edge and emerging trends with lots of practical experience will make the learning more interesting and stimulating.

4. Learning Outcomes

The core philosophy of the syllabus is to (i) give strong foundation on core Computer Science subjects;(ii) expose the student to emerging trends in a gradual and incremental way; (iii) offer specialization on a chosen area (iv) Create a research temper among students in the whole process; (v) prepare student community for the demands of ICT industry.

5. Input from stakeholders with relevant information

Introduction of new subjects/concepts in courses to furnish students with the skills needed to contribute in an ever-evolving IT field. Inputs for industrial, academic and research experts have shaped the syllabus to be extensive and comprehensive.

Structure of the Syllabus

This is the syllabus for the semester–III and semester–IV of MSc Computer Science program of HSNC University, Mumbai to be implemented from the year 2021-2022.

Semester – III

Course Code	Course Nomenclature	Lecture in Hours	Credits
PS-SCS-301	Design and Implementation of Modern Compilers	60	4
PS-SCS-302	Social Network Analysis	60	4
PS-SCS-303	Elective I: Course from Track A	60	4
PS-SCS-304	Elective I: Course from Track B		
PS-SCS-305	Elective II: Course from Track C	60	4
PS-SCS-306	Elective II: Course from Track D		
PS-SCS-3P1	Practical based on PS-SCS-301 and PS-SCS-302	60	4
PS-SCS-3P2	Practical based on Elective I and Elective II	120 (60+60)	4

Semester – IV

Course Code	Course Nomenclature	Lecture in Hours	Credits
PS-SCS-401	Simulation and Modeling	60	4
PS-SCS-402	Specialization: Course from Track A		

PS-SCS-403	Specialization: Course from Track B	60	4
PS-SCS-404	Specialization: Course from Track C		
PS-SCS-405	Specialization: Course from Track D		
PS-SCS-4P1	Practical based on PS-SCS-401 and Specialization Track.	120 (60+60)	4
PS-SCS-4P2	Internship with Industry	-	6
PS-SCS-4P3	Project Dissertation	-	6

Track A	Artificial Intelligence and Machine Learning
Track B	Cloud Computing
Track C	Big Data Engineering
Track D	Human Computer Interaction

Track	Semester III	Semester IV
Track A	Deep Learning	Natural Language Processing
Track B	Cloud Application Development	Cloud Computing – Google Cloud Platform
Track C	Introduction to Big Data Analytics and No SQL Databases	Big Data Analytics: Tools and Frameworks
Track D	Advanced Human Computer Interaction	User Experience Design

A student is expected to continue with the same tracks he or she has taken in semester-II for elective –I and elective –II. Each of these theory courses (compulsory as well as elective) is of four credits each and is expected to complete in 60 hours.

Part 2 -Second Year Semester III Internal and External Detailed Evaluation Scheme

Sr. No.	Subject Code	Subject Title	Periods Per Week					Credits	Internals (40)				Total Marks
			Units	S.L.	L	T	P		SLE	CT + AT = 15 + 5	PA	SEE	
1	PS-SCS-301	Design and Implementation of Modern Compilers	4	20% *	4	0	-	4	10	20	10	60	100
2	PS-SCS-302	Social Network Analysis	4	20% *	4	0	-	4	10	20	10	60	100
3	PS-SCS-303	Elective I: Course from Track A	4	20% *	4	0	-	4	10	20	10	60	100
	PS-SCS-304	Elective I: Course from Track B											
4	PS-SCS-305	Elective II: Course from Track C	4	20% *	4	0	-	4	10	20	10	60	100
	PS-SCS-306	Elective II: Course from Track D											

5	PS-SCS-3P1	Practical based on PS-SCS-301 & PS-SCS-302					6	4				100 (80+20)	100
6	PS-SCS-3P2	Practical based on Elective I and II					6	4				100 (80+20)	100
	Total Lectures / Credits							24	Total Marks				600

L: Lecture: Tutorials, P: Practical Ct-Core Theory, Cp-Core Practical, SLE- Self learning evaluation CT-Commutative Test, SEE- Semester End Examination, PA-Project Assessment, AT- Attendance

***One to two lectures to be taken for CONTINUOUS self -learning Evaluation.**

Part 3 - Detailed Syllabus of Semester – III

Course Code: PS-SCS-301 Design and implementation of Modern Compilers

Unit	Content	No. of Lectures
1	**Unit I: Introduction to Compilers** The structure of a compiler, A simple approach to the design of lexical analyzers, Regular expressions, Finite automata, from regular expressions to finite automata, Minimizing the number of states of a DFA, Context-free grammars, Derivations and Parse trees, Parsers, Shift-reduce parsing, Operator-precedence parsing, Top- down parsing, Predictive parsers.	**15**
2	**Unit II: 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, Constructing LALR sets of items	**15**
3	**Unit III: Advanced syntax analysis and basic semantic analysis** Syntax-directed translation schemes, Implementation of syntax-directed translators, Initial introduction to the ongoing Tiger compiler, bindings for the Tiger compiler, typechecking expressions, type-checking declarations, activation records, stack frames, frames in the Tiger compiler, translation to intermediate code, intermediate representation trees, translation into trees, declarations, basic blocks and traces, taming conditional branches, liveness analysis, solution of dataflow equations, liveness in the Tiger compiler, interference graph construction.	**15**
4	**Unit IV: Dataflow analysis and loop optimization** The principle sources of optimization, Loop optimization: The DAG representation of basic blocks, Dominators, Reducible flow graphs, Depth-first search, Loop-invariant computations, Induction variable elimination, Some other loop optimizations. Dataflow Analysis: intermediate representation for flow analysis, various dataflow analyses, transformations using dataflow analysis, speeding up dataflow analysis, alias analysis.	**15**

Text Book

1. Compilers: Principles, Techniques and Tools 2nd edition, Alfred V. Aho , Monica S. Lam , Ravi Sethi , Jeffrey D. Ullman , Pearson (2011)
2. Modern Compiler Implementation in Java, Second Edition, Andrew Appel and Jens Palsberg, Cambridge University Press (2004).

References

1. Principles of Compiler Design, Alfred Aho and Jeffrey D. Ullman, Addison Wesley (1997).
2. Compiler design in C, Allen Holub, Prentice Hall (1990).

Course Code: PS-SCS-302 Social Network Analysis

Unit	Content	No. of Lectures
1	Unit I: Introduction to social network analysis (SNA) Introduction to networks and relations- analyzing relationships to understand people and groups, binary and valued relationships, symmetric and asymmetric relationships, multimode relationships, Using graph theory for social networks analysis- adjacency matrices, edge-lists, adjacency lists, graph traversals and distances, depth-first traversal, breadth-first traversal paths and walks, Dijkstra's algorithm, graph distance and graph diameter, social networks vs. link analysis, ego-centric and socio-centric density.	**15**
2	Unit II: Networks, Centrality and centralization in SNA Understanding networks- density, reachability, connectivity, reciprocity, group-external and group-internal ties in networks, ego networks, extracting and visualizing ego networks, structural holes, Centrality- degree of centrality, closeness and betweenness centrality, local and global centrality, centralization and graph centers, notion of importance within network, Google pagerank algorithm, Analyzing network structurebottom-up approaches using cliques, N-cliques, N-clans, K-plexes, K-cores, F-groups and top-down approaches using components, blocks and cut-points, lambda sets and bridges, and factions.	**15**
3	Unit III: Measures of similarity and structural equivalence in SNA Approaches to network positions and social roles- defining equivalence or similarity, structural equivalence, automorphic equivalence, finding equivalence sets, brute force and Tabu search, regular equivalence, equivalence of distances: Maxsim, regular equivalence, Measuring similarity/dissimilarity- valued relations, Pearson correlations covariance and cross-products, Understanding clustering-agglomerative and divisive clusters, Euclidean, Manhattan, and squared distances, binary relations, matches: exact, Jaccard, Hamming,	**15**
4	Unit IV: Two-mode networks for SNA Understanding mode networks- Bi-partite data structures, visualizing two-mode data, quantitative analysis using two-mode Singular value decomposition (SVD) analysis, two-mode factor analysis, two-mode correspondence analysis, qualitative analysis using two-mode core-periphery analysis, two-mode factions analysis, affiliation and attribute networks.	**15**

Text Book

1. Introduction to Social Network Methods: Robert A. Hanneman, Mark Riddle, University of California, 2005 [Published in digital form and available at http://faculty.ucr.edu/~hanneman/nettext/index.html].
2. Social Network Analysis for Startups- Finding connections on the social web: Maksim Tsvetovat, Alexander Kouznetsov, O'Reilly Media, 2011.
3. Social Network Analysis- 3rd edition, John Scott, SAGE Publications, 2012.

References

1. Exploratory Social Network Analysis with Pajek, Second edition: Wouter de Nooy, Andrej Mrvar, Vladimir Batagelj, Cambridge University Press, 2011.
2. Analyzing Social Networks, Stephen P Borgatti, Martin G. Everett, Jeffrey C. Johnson, SAGE Publications, 2013.
3. Statistical Analysis of Network Data with R: Eric D. Kolaczyk, Gábor Csárdi, Springer, 2014.
4. Network Analysis: Methodological Foundations, (Editors) Ulrik Brandes, Thomas Erlebach. Springer, 2005.
5. Models and Methods in Social Network Analysis: (Editors) Peter J. Carrington, John Scott, Stanley Wasserman, Cambridge University Press, 2005.

Course Code: PS-SCS-303

Elective I: Track A: Artificial Intelligence and Machine Learning

Deep Learning

Unit	Content	No. of Lectures
1	**1.1** The Neural Network: Building Intelligent Machines, The Limits of Traditional Computer Programs, The Mechanics of Machine Learning, The Neuron, Expressing Linear Perceptrons as Neurons, Feed-Forward Neural Networks, Linear Neurons and Their Limitations, Sigmoid, Tanh, and ReLU Neurons, Softmax Output Layers. **1.2** Training Feed-Forward Neural Networks: The Fast-Food Problem, Gradient Descent, The Delta Rule and Learning Rates, Gradient Descent with Sigmoidal Neurons, The Backpropagation Algorithm.	**15**
2	**1.1** Optimization for Training Deep Models: Stochastic Gradient Descent, Momentum, Nesterov Momentum, AdaGrad, RMSProp, Adam. **1.2** Regularization for Deep Learning: Bias and Variance, Overfitting in deep neural networks, L2 regularization, Dataset Augmentation, Early Stopping.	**15**

3	**3.1** Convolutional Neural Networks, LeNet, AlexNet, ZF-Net, VGGNet, GoogLeNet, ResNet, Visualizing Convolutional Neural Networks **3.2** Batch Normalization, Ensemble Methods, Dropout, Hyperparameter Tuning	**15**
4	**4.1** Recurrent Neural Networks, Backpropagation through time (BPTT), Vanishing and Exploding Gradients, Truncated BPTT, GRU, LSTMs **4.2** Encoder Decoder Models, Attention Mechanism, Attention over images	**15**

References:

1. Deep Learning, An MIT Press book, Ian Goodfellow and Yoshua Bengio and Aaron Courville http://www.deeplearningbook.org
2. Fundamentals of Deep Learning: Designing Next-Generation Machine Intelligence Algorithms, Nicholas Locascio and Nikhil Buduma, O'Reilly, 1st edition, 2017.
3. Hands-on Machine Learning with Scikit-Learn, Keras and TensorFlow – Concepts, Tools and Techniques to build Intelligent Systems, Aurelien Geron.
4. Deep Learning CookBook, Douwe Osinga, Douwe Osinga, 1st edition, 2017.
5. https://onlinecourses.nptel.ac.in/noc21_cs76/preview

Course Code: PS-SCS-304

Elective I: Track B: Cloud Computing

Cloud Application Development

Unit	Content	No. of Lectures
1	**Implementing Microservices:** Client to microservices communication, Interservice communication, data considerations, security, monitoring, microservices hosting platform options. **Azure Service Fabric:** Introduction, core concepts, supported programming models, service fabric clusters, develop and deploy applications of service fabric. **Monitoring Azure Service Fabric Clusters:** Azure application, resource manager template, Adding Application Monitoring to a Stateless Service Using Application Insights, Cluster monitoring, Infrastructure monitoring. **Azure Kubernetes Service (AKS):** Introduction to kubernetes and AKS, AKS development tools, Deploy applications on AKS.	**15**

2	**Monitoring AKS:** Monitoring, Azure monitor and analytics, monitoring AKS clusters, native kubernetes dashboard, Prometheus and Grafana. **Securing Microservices:** Authentication in microservices, Implenting security using API gateway pattern, Creating application using Ocrlot and securing APIs with Azure AD. **Database Design for Microservices:** Data stores, monolithic approach, Microservices approach, harnessing cloud computing, dataase options on MS Azure, overcoming application development challenges. **Building Microservices on Azure Stack:** Azure stack, Offering IaaS, PaaS on-premises simplified, SaaS on Azure stack.	**15**
3	**.NET DevOps for Azure:** DevOps introduction, Problem and solution. **Professional Grade DevOps Environment:** The state of DevOps, professional grade DevOps vision, DevOps architecture, tools for professional DevOps environment, DevOps centered application. **Tracking work:** Process template, Types of work items, Customizing the process, Working with the process. **Tracking code:** Number of repositories, Git repository, structure, branching pattern, Azure repos configuration, Git and Azure. **Building the code:** Structure of build, using builds with .NET core and Azure pipelines, **Validating the code:** Strategy for defect detection, Implementing defect detection. **Release candidate creation:** Designing release candidate architecture, Azure artifacts workflow for release candidates,	**15**
4	**Deploying the release:** Designing deployment pipeline, Implementing deployment in Azure pipelines. **Operating and monitoring release:** Principles, Architectures for observability, Jumpstarting observability. **Introduction to APIs:** Introduction, API economy, APIs in public sector. **API Strategy and Architecture:** API Strategy, API value chain, API architecture, API management. **API Development:** Considerations, Standards, kick-start API development, team orientation.	**15**

	API Gateways: API Gateways in public cloud, Azure API management, AWS API gateway. **API Security:** Request-based security, Authentication and authorization.	

References:

1. Building Microservices Applications on Microsoft Azure- Designing, Developing, Deploying, and Monitoring, Harsh Chawla Hemant Kathuria, Apress, 2019.
2. .NET DevOps for Azure A Developer's Guide to DevOps Architecture the Right Way, Jeffrey Palermo, Apress, 2019.
3. Practical API Architecture and Development with Azure and AWS - Design and Implementation of APIs for the Cloud, Thurupathan Vijayakumar, Apress, 2018.

Course Code: PS-SCS-305

Elective II: Track C: Big Data Engineering

Introduction to Big Data Analytics and NoSQL Databases

Unit	Content	No. of Lectures
1	**1.1 Big Data Analytics:** The data ecosystem, Components of data ecosystem, Data categorization, Sources of data, Overview of data repositories, Introduction to Big Data, Big Data Characteristics, Types of Big Data, Traditional Versus Big Data Approach, Big Data Customer Scenarios, Limitations and Solutions of Existing Data, Analytics Architecture with Uber Use Case, Technologies Available for Big Data, Infrastructure for Big Data, Use of Data Analytics, Big Data Challenges, Desired Properties of a Big Data System, Case Study of Big Data Solutions, How Hadoop Solves the Big Data Problem? **1.2 Hadoop:** What is Hadoop? Hadoop's Key Characteristics, Hadoop Ecosystem and HDFS Hadoop Core Components, Rack Awareness and Block Replication YARN and its Advantage, Hadoop Cluster and its Architecture Hadoop: Different Cluster Modes, Big Data Analytics with Batch & Real-time Processing.	**15**
2	**Learning the NoSQL Basics** **2.1 Introduction to NoSQL:** Characteristics of NoSQL, NoSQL Storage types, Advantages and Drawbacks, NoSQL Products **2.2 Interfacing and interacting with NoSQL:** Storing Data In and Accessing Data from MongoDB, Redis, HBase and Apache Cassandra, Language Bindings for NoSQL Data Stores **2.3 Understanding the storage architecture:** Working with Column-Oriented Databases, HBase Distributed Storage Architecture, Document Store Internals,	**15**

	Understanding Key/Value Stores in Memcached and Redis, Eventually Consistent Non-relational Databases **2.4 Performing CRUD operations:** Creating Records, Accessing Data, Updating and Deleting Data	
3	**Gaining Proficiency with NoSQL** **3.1 Querying NoSQL Stores:** Similarities Between SQL and MongoDB Query Features, Accessing Data from Column-Oriented Databases Like HBase, Querying Redis Data Stores **3.2 Indexing And Ordering Data Sets:** Essential Concepts behind a Database Index, Indexing and Ordering in MongoDB, couchDB and Apache Cassandra **3.3 Managing Transactions and Data Integrity:** RDBMS and ACID, Distributed ACID Systems, Upholding CAP, Consistency Implementations **3.4 Using NoSQL in the Cloud:** Google App Engine Data Store, Amazon SimpleDB	**15**
4	**Map Reduce** **4.1 Introduction to Map Reduce:** The map tasks, Grouping by key, The reduce tasks, Combiners, Details of MapReduce Execution, Coping with node failures. **4.2 Algorithms Using MapReduce:** Matrix-Vector Multiplication, Computing Selections and Projections, Union, Intersection, and Difference, Natural Join. **4.3 Extensions to MapReduce:** Workflow Systems, Recursive extensions to MapReduce, Common map reduce algorithms.	**15**

Text Book:

1. Professional NoSQL By Shashank Tiwari, Wrox-John Wiley & Sons, Inc, 2011
2. Getting Started with NoSQL, Gaurav Vaish, Packt Publishing Ltd, 2013

References:

1. SQL & NoSQL Databases, Andreas Meier · Michael Kaufmann, Springer Vieweg, 2019
2. Practical Hadoop Migration: How to Integrate Your RDBMS with the Hadoop Ecosystem and Re-Architect Relational Applications to NoSQL By Bhushan Lakhe, Apress; 1st edition, 2016.

Course Code: PS-SCS-306

Elective II: Track C: Human Computer Interaction

Advanced Human Computer Interaction

Unit	Content	No. of Lectures
1	**The Interaction:** Models of interaction, Design Focus, Frameworks and HCI, Ergonomics, Interaction styles, Elements of the WIMP interface, Interactivity **Paradigms:** Introduction, Paradigms for interaction **Interaction design basics:** What is design?, The process of design, User focus, Cultural probes, Navigation design, the big button trap, Modes, Screen design and layout, Alignment and layout matters, Checking screen colors, Iteration and prototyping **HCI in the software process**: The software life cycle, Usability engineering , Iterative design and prototyping, Prototyping in practice, Design rationale **Design**: Principles to support usability, Standards, Guidelines, Golden rules and heuristics, HCI patterns	**15**
2	**Implementation support:** Elements of windowing systems, Programming the application, Going with the grain, Using toolkits, User interface management systems **Evaluation techniques:** What is evaluation?, Goals of evaluation, Evaluation through expert analysis, Evaluation through user participation, Choosing an evaluation method **Universal design:** Universal design principles, Multi modal interaction, Designing websites for screen readers, Choosing the right kind of speech, Designing for diversity	**15**
3	**User support:** Requirements of user support, Approaches to user support, Adaptive help systems, Designing user support systems **Cognitive models:** Goal and task hierarchies, Linguistic models, The challenge of display-based systems, Physical and device models, Cognitive architectures **Socio-organizational issues and stakeholder requirements:** Organizational issues, Capturing requirements **Communication and collaboration models:** Face-toface communication, Conversation, Text-based communication, Group working	**15**
4	**Task analysis:** Differences between task analysis and	**15**

	other techniques, Task decomposition, Knowledge based analysis, Entity relationship-based techniques, Sources of information and data collection, Uses of task analysis **Dialog notations and design:** What is dialog?, Dialog design notations, Diagrammatic notations, Textual dialog notations, Dialog semantics, Dialog analysis and design **Models of the system:** Standard formalisms, Interaction models, Continuous behavior **Modeling rich interaction:** Status–event analysis, Rich contexts, Low intention and sensor-based interaction	

References:

1. Human Computer Interaction, Alan Dix, Janet Finlay, Gregory Abowd, Russell Beale, Pearson Education
2. Designing the User Interface, Shneiderman B., Plaisant C., Cohen M., Jacobs S.

Part 4 – Detailed Scheme Practical

Course Code: **PS-SCS-3P1**	**PS-SCS-301 & PS-SCS-302**	Total Credits: **4**	No. of Lectures: **120**

PS-SCS-301: Design and Implementation of Modern Compilers

1. Write a program to convert the given NDFA to DFA.
2. Write a program to convert the given Right Linear Grammar to Left Linear
3. Grammar form.
4. Write a program to illustrate the generation on SPM for the input grammar.
5. Implement a simple program analyzer and interpreter for the straight-line programming language.
6. Add semantic actions to your parser to produce abstract syntax for the MiniJava language together with a PrettyPrintVisitor.
7. Design a set of visitors, which translate a MiniJava program into intermediate representation trees.
8. Implement the translation to Assem instructions for your favorite instruction set (let μ stand for Sparc, Mips, Alpha, Pentium, etc.) Using maximal munch.
9. Write a code to generate the DAG for the input arithmetic expression.
10. Write a program to demonstrate loop unrolling and loop splitting for the given code sequence containing loop.

PS-SCS-302: Social Network Analysis

1. Write a program to compute the following for a given a network: (i) number of edges, (ii) number of nodes; (iii) degree of node; (iv) node with lowest degree; (v) the adjacency list; (vi) matrix of the graph.
2. Perform following tasks: (i) View data collection forms and/or import onemode/two-mode datasets; (ii) Basic Networks matrices transformations

3. Compute the following node level measures: (i) Density; (ii) Degree; (iii) Reciprocity; (iv) Transitivity; (v) Centralization; (vi) Clustering.
4. For a given network find the following: (i) Length of the shortest path from a given node to another node; (ii) the density of the graph; (iii) Draw egocentric network of node G with chosen configuration parameters.
5. Write a program to distinguish between a network as a matrix, a network as an edge list, and a network as a sociogram (or "network graph") using 3 distinct networks representatives of each.
6. Write a program to exhibit structural equivalence, automatic equivalence, and regular equivalence from a network.
7. Create sociograms for the persons-by-persons network and the committee-by-committee network for a given relevant problem. Create one-mode network and two-node network for the same.
8. Perform SVD analysis of a network.
9. Identify ties within the network using two-mode core periphery analysis.
10. Find "factions" in the network using two-mode faction analysis.

Note:
One may use programming languages like R, Python, Pajek etc and open software/tools like (i) EGONet; (ii) Ora; (iii) Netlogo; (iv) Pajek; and (v) NetDraw; to do the practical experiments.

Course Code: **PS-SCS-3P2**	**Practical based on Elective I and Elective II**	Total Credits: **4**	No. of Lectures: **120**
Elective I: PS-SCS-303: Deep Learning 8 – 10 Practicals based on the syllabus topics.			
Elective I: PS-SCS-304: Cloud Application Development 8 – 10 Practicals based on the syllabus topics.			
Elective II: PS-SCS-305: Introduction to Big Data Analytics and NoSQL Databases 8 – 10 Practicals based on the syllabus topics.			
Elective II: PS-SCS-306: Advanced Human Computer Interaction 8 – 10 Practicals based on the syllabus topics.			

Part 5 - Second Year Semester -IV Internal and External Detailed Evaluation Scheme

Sr. No.	Subject Code	Subject Title	Periods Per Week					Credits	Internals (40)				Total Marks
			Units	S.L.	L	T	P		SLE	CT + AT = 15 + 5	PA	SEE	
1	PS-SCS-401	Simulation and Modeling	4	20% *	4	0	-	4	10	20	10	60	100
2	PS-SCS-402	Specialization : Course from Track A	4	20% *	4	0	-	4	10	20	10	60	100
	PS-SCS-403	Specialization : Course from Track B											
	PS-SCS-404	Specialization : Course from Track C											
	PS-SCS-405	Specialization : Course from Track D											
3	PS-SCS-4P1	Practical based on PS-SCS-401 and Specialization Track	-	-	-	-	6	4	-	-	-	80+20	100

4	PS-SCS-4P2	Internship with Industry	-	-	-	-		6					150
5	PS-SCS-4P3	Project Dissertation	-	-	-	-		6					150
	Total Lectures / Credits							24	Total Marks				600

Internship with Industry

The syllabus proposes an internship for about 8 weeks to 12 weeks to be done by a student. It is expected that a student chooses an IT or IT-related industry and formally works as a full-time intern during the period. The student should subject oneself with an internship evaluation with proper documentation of the attendance and the type of work he or she has done in the chosen organization. Proper certification (as per the guidelines given in Appendix 1 and 2) by the person, to whom the student was reporting, with Organization's seal should be attached as part of the documentation.

Project Dissertation

The syllabus proposes project implementation as part of the semester–IV. The project implementation is continuation of the project proposal the students has submitted and evaluated in semester-III. The student is expected to continue with the proposal made and examined in the semester-III and implement the same in the semester–IV. In addition, experimental set up, analysis of results, comparison with results of related works, conclusion and future prospects will be part of the project implementation. A student is expected to make a project implementation report and appear for a project viva. He or she needs to spend around 200 hours for the project implementation, which fetches 6 credits.

Part 6 - Detailed Syllabus of Semester – IV

PS-SCS-401: Simulation and Modeling

Unit	Content	No. of Lectures
1	Unit I: Introduction to Simulation, Need of Simulation, Time to simulate, Inside simulation software: Modeling the progress of Time, Modeling Variability, Conceptual Modeling: Introduction to Conceptual modeling, Defining conceptual model, Requirements of the conceptual model, Communicating the conceptual	**15**

	model, Developing the Conceptual Model: Introduction, A framework for conceptual modeling, methods of model simplification.	
2	Unit II: Model Verification and Validation Data Collection and Analysis: Introduction, Data requirements, Obtaining data, Representing unpredictable variability, Selecting statistical distributions. Obtaining Accurate Results: Introduction, The nature of simulation models and simulation output, Issues in obtaining accurate simulation results, example model, dealing with initialization bias: warm-up and initial conditions, Selecting the number of replications and run-length. Searching the Solution Space: Introduction, The nature of simulation experimentation, Analysis of results from a single scenario, Comparing alternatives, Search experimentation, and Sensitive analysis. Verification, Validation and Confidence: Introduction, Defining Verification and Validation, The difficulties of verification and validation, Methods of verification and validation, Independent verification and validation.	**15**
3	Unit III: Modeling and simulation modeling Types of models, Analytical vs Simulation modeling, Application of simulation modeling, Level of abstraction, Simulation Modeling. Methods, System Dynamics, Discrete Event Modeling, Agent Based modeling: Introduction to Agent, Agent-based modeling, Time in agent based models, Space in agent based models, Discrete space, Continuous space movement in continuous space, Communication between agents, Dynamic creation and destruction of agents, Statics on agent population, Condition triggered events and transition in agents. Building agents based models: The problem statement, Phases of modeling, Assumptions, 3 D animation. Dynamics Systems: Stock and flow diagrams, examples of stock and flow diagrams. Multi-method modeling: Architecture, Technical aspects of combining modeling methods, Examples.	**15**
4	Unit IV: Design and behavior of models Designing state-based behavior: Statecharts, State transitions, Viewing and debugging Statecharts at runtime, Statecharts for dynamic objects. Discrete events and Event model object: Discrete event, Event-the simplest low level model object, Dynamic events, and Exchanging data with external world. Presentation and animation: Working with shapes, groups and colors, Designing interactive models: using controls, Dynamic properties of controls, 3D Animation. Randomness in Models: Probability distributions, sources of randomness in the model, randomness in system dynamics model, random number generators, Model time, date and calendar: Virtual and real time: The model time, date and calendar, Virtual and real-time execution modes.	**15**

Text Book:

1. Simulation: The Practice of Model Development and Use by Stewart Robinson, John Wiley and Sons, Ltd, 2004.

2. The Big Book of Simulation Modeling: Multi Method Modeling by Andrei Borshchev, 2013.

References:

1. Agent Based Modeling and Simulation, Taylor S, 2014.
2. Simulation Modeling Handbook: A Practical Approach, Christopher A. Chung, 2003.
3. Object Oriented Simulation: A Modeling and Programming Perspective, Garrido, José M, 2009.
4. Simulation, Modeling and Analysis, Averill M Law and W. David Kelton, "Tata McGraw Hill, Third Edition, 2003.
5. Process Control: Modeling, Design and Simulation, Wayne Bequette W, Prentice Hall of India, 2003.

Course Code: PS-SCS-402

Specialization: Track A: Artificial Intelligence and Machine Learning

(Natural Language Processing)

Unit	Content	No. of Lectures
1	**Introduction to Natural Language Processing (NLP) and Language Modelling:** **Introduction to NLP:** Introduction and applications, NLP phases, Difficulty of NLP including ambiguity; Spelling error and Noisy Channel Model; Concepts of Parts-of speech and Formal Grammar of English. **Language Modelling:** N-gram and Neural Language Models Language Modelling with N-gram, Simple N-gram models, smoothing (basic techniques), Evaluating language models; Neural Network basics, Training; Neural Language Model, Case study: application of neural language model in NLP system development **Python Libraries for NLP:** Using Python libraries/packages such as NaturalLanguage Toolkit (NLTK), spaCy, genism	**15**
2	Morphology & Parsing in NLP Computational morphology & Parts-of-speech Tagging: basic concepts; Tagset; Lemmatization, Early approaches: Rule-based and TBL; POS tagging using HMM, Introduction to POS Tagging using Neural Model. Parsing Basic concepts: top-down and bottom-up parsing, treebank; Syntactic parsing: CKY parsing; Statistical Parsing basics: Probabilistic Context-Free Grammar (PCFG); Probabilistic CKY	**15**

	Parsing of PCFGs.	
3	Semantics and Word Embedding Semantics Vector Semantics: Words and Vector; Measuring Similarity; Semantics with dense vectors; SVD and Latent Semantic Analysis Embeddings from prediction: Skip-gram and Continuous Bag of words; Concept of Word Sense; Introduction to WordNet	**15**
4	NLP Applications and Case Studies Intelligent Work Processors: Machine Translation; User Interfaces; man-machine Interfaces: Natural language Querying Tutoring and Authoring Systems. Speech Recognition Commercial use of NLP: NLP in customer Service, Sentiment Analysis, Emotion Mining, Handling Frauds and SMS, Bots, LSTM & BERT models, Conversations	**15**

TEXT BOOK:

1. Speech and Language Processing‖, Jurafsky Dan and Martin James H., 3rd Edition, Pearson, 2018.
2. Natural Language Processing with Python‖, Steven Bird, Ewan Klein, and Edward Loper, 2nd Edition, O'Reilly, 2016.

REFERENCE BOOKS:

1. Practical Natural Language Processing with Python‖, Mathangi Sri, Apress, 2021
2. "Handbook of Computational Linguistics and Natural Language Processing‖, Martin Whitehead, Clanrye International, 2020
3. Handbook of Natural Language Processing‖, Nitin Indurkhya, and Fred J. Damerau, Pearson; 2nd edition, 2008
4. Foundations of Statistical Natural Language Processing‖, Manning, Christopher and Heinrich, Schutze, MIT Press, 1997

Course Code: PS-SCS-403

Specialization: Track B: Cloud Computing

(Cloud Computing – Google Cloud Platform)

Unit	Content	No. of Lectures
1	Cloud Computing, Cloud vs Traditional Architecture, IaaS, PaaS and SaaS, Google Architecture GCP Console, Understanding Projects, Billing in GCP, Install and Configure Cloud SDK, Use Cloud Shell Lab: A tour of Qwiklabs and Google Cloud Platform Lab: Getting Started with Cloud Shell and gcloud GCP APIs, Cloud Console Mobile App	**15**
2	Use GCP to build your apps, Compute Options in the Cloud, Exploring IaaS with Compute Engine Lab: Creating a Virtual Machine Configuring Elastic Apps with Autoscaling, Exploring PaaS with App Engine Lab: App Engine: Qwik Start - Python Event-Driven Programs with Cloud Functions Lab: Cloud Functions: Qwik Start - Command Line Containerizing and Orchestrating Apps with GKE Storage Options in the Cloud, Structured and Unstructured Storage in the Cloud Lab: Cloud Storage: Qwik Start - CLI/SDK SQL Managed Services, Exploring Cloud SQL Lab: Loading Data into Cloud SQL Cloud Spanner as a Managed Service, NoSQL Managed Services Options, Cloud Datastore a NoSQL Document Store Lab: App Dev: Storing Application Data in Cloud Datastore - Python	**15**

	Cloud Bigtable as a NoSQL Option	
3	Introduction to API, The Purpose of APIs, Cloud Endpoints Lab: Cloud Endpoints: Qwik Start Using Apigee Edge, Managed Message Services, Cloud Pub/Sub Lab: Cloud Pub/Sub: Qwik Start - Python Introduction to Security in the Cloud, The Shared Security Model, Encryption Options, Authentication and Authorization with Cloud IAM Lab: User Authentication: Cloud Identity-Aware Proxy Identify Best Prcatices for Authorization using Cloud IAM	**15**
4	Introduction to Networking in the Cloud, Defining a Virtual Private Cloud, Public and Private IP Address Basics, Googles Network Architecture, Routes and Firewall Rules in the Cloud Lab: VPC Networking Fundamentals Multiple VPC Networks (with Lab) Lab: VPC Networks - Controlling Access Building Hybrid Clouds using VPNs, Interconnecting and Direct Peering, Different Options for Load Balancing Lab: HTTP Load Balancer with Cloud Armor Lab: Create an Internal Load Balancer **Introduction to ML, ML and GCP, Building Bespoke ML models, Cloud AutoML** Lab: Classify Images of CLouds in the Cloud with AutoML Vision Googles Pre-trained ML APIs Qwik Start Labs: Cloud Natural Language API, Cloud Search API, Video Intelligence API	**15**

References:

https://cloud.google.com/edu/curriculum

https://nptel.ac.in/courses/106/105/106105223/

Course Code: PS-SCS-404

Specialization: Track C: Big Data Engineering

(Big Data Analytics: Tools and Frameworks)

Unit	Content	No. of Lectures
1	Data Product, Building Data Products at Scale with Hadoop, Data Science Pipeline and Hadoop Ecosystem, Operating System for Big Data, Concepts, Hadoop Architecture, Working with Distributed file system, Working with Distributed Computation, Framework for Python and Hadoop Streaming, Hadoop Streaming, MapReduce with Python, Advanced MapReduce. In- Memory Computing with Spark, Spark Basics, Interactive Spark with PySpark, Writing Spark Applications	**15**
2	Distributed Analysis and Patterns, Computing with Keys, Design Patterns, Last-Mile Analytics, Data Mining and Warehousing, Structured Data Queries with Hive, HBase, Data Ingestion, Importing Relational data with Sqoop, Injesting stream data with flume. Analytics with higher level APIs, Pig, Spark's higher level APIs.	**15**
3	What is Scala? Why Scala for Spark? Scala in other Frameworks Introduction to Scala REPL Basic Scala Operations Variable Types in Scala Control Structures in Scala Foreach loop, Functions and Procedures Collections in Scala- Array ArrayBuffer, Map, Tuples, Lists. Functional Programming Higher Order Functions Anonymous Functions Class in Scala Getters and Setters Custom Getters and Setters Properties with only Getters Auxiliary Constructor and Primary Constructor Singletons Extending a Class Overriding Methods Traits as Interfaces and Layered Traits	**15**
4	Spark SQL, DataFrames and Datasets Spark Machine Learning Library: MLlib	**15**

References:

1. Big Data and Analytics, Seema Acharya and Subhashini Chellappan, Wiley, First Edition.
2. Data Analytics with Hadoop An Introduction for Data Scientists, Benjamin Bengfort and
3. Jenny Kim, O 'Reilly, 2016.
4. Big Data and Hadoop, V.K Jain, Khanna Publishing, First Edition, 2018.
5. Apache Spark: https://spark.apache.org/docs/latest/

Course Code: PS-SCS-405

Specialization: Track D: Human Computer Interaction

(User Experience Design)

Unit	Content	No. of Lectures
1	HCI and Usability, Usability Paradigms and Usability Principles, User Interface Design Tools Introduction: Introduction to interface design, Understanding and conceptualizing Interface, Understanding user's conceptual cognition. What is UX, Ubiquitous interaction, Emerging desire for usability, From usability to user experience, Emotional impact as part of the user experience, User experience needs a business case, Roots of usability. Elements of UX Design: Core Elements of User Experience, Working of UX elements	**15**
2	The UX Design Process – Understanding Users: Defining the UX, Design Process and Methodology, Understanding user requirements and goals, Understanding the Business Requirements/Goals, User research, mental models, wireframes, prototyping, usability testing. The UX Design Process- The Structure: Information Architecture and Interaction Design: Visual Design Principles ,Information Design and Data Visualization Interaction Design ,Information Architecture , Wire framing & Storyboarding, UI Elements and Widgets, Screen Design and Layouts	**15**
3	UX Design Process- Prototype and Test: Testing your Design, Usability Testing, Types of Usability Testing ,Usability Testing Process, Preparing and planning for the Usability Tests, Prototype your Design to Test, Introduction of prototyping tools, conducting Usability Test, communicating Usability Test Results	**15**
4	UX Design Process- Iterate/ Improve and Deliver: Understanding the Usability Test, findings, Applying the Usability Test, feedback in improving the design.	**15**

	Communication with implementation team. UX Deliverables to be given to implementation team UX Design Guidelines: Introduction, Using and interpreting design guidelines, Human memory limitations, Selected UX design guidelines and examples, Planning, Translation, Physical actions, Outcomes, Assessment, Overall.	

Textbooks:

1. The UX Book by Rex Hartson and Pardha Pyla

2. Smashing UX Design by Jesmond Allen and James Chudley

3. Lean UX: Applying Lean Principles to Improve User Experience by Jeff Gothelf and Josh Seiden

4. Don’t Make Me Think, Revisited by Steve Krug

5. The User Experience Team of One by Leah Buley

6. The Elements of User Experience by Jesse James Garrett

7. Sketching User Experiences: The Workbook by Saul Greenberg, Sheelagh Carpendale, Nicolai Marquardt and Bill Buxton

8. Interaction Design, Beyond Human Computer Interaction, Rogers, Sharp, Preece Wiley

India Pvt Ltd.

9. The essentials of Interaction Design, Alan Cooper, Robert Reimann, David Cronin

10. Designing The user Interface by Shneiderman, Plaisant,Cohen,Jacobs Pearson

References:

1. A Project Guide to UX Design by Russ Unger and Carolyn Chandler

2. Agile Experience Design by Lindsay Ratcliffe and Marc McNeill

3. Universal Principles of Design by William Lidwell, Kritina Holden and Jill Butler

4. Human Computer Interaction by Alan Dix

5. Observing the User Experience: A Practitioner's Guide to User Research by Mike Kuniavsky

Part 6 – Detailed Scheme Practical

Course Code: **PS-SCS-4P1**	**PS-SCS-401 & Specialization Track**	Total Credits: **4**	No. of Lectures: **120**

PS-SCS-401: Simulation and Modeling

Design and develop agent based model by
- Creating the agent population
- Defining the agent behavior
- Add a chart to visualize the model output.

[Use a case scenario like grocery store, telephone call center etc for the purpose].

Design and develop agent based model by
- Creating the agent population
- Defining the agent behavior
- Adding a chart to visualize the model output
- Adding word of mouth effect
- Considering product discards

Considering delivery time
[Use a case scenario like restaurant].

Design and develop agent based model by
- Creating the agent population
- Defining the agent behavior
- Adding a chart to visualize the model output
- Adding word of mouth effect
- Considering product discards
- Consider delivery time
- Simulating agent impatience
- Comparing model runs with different parameter values

[Use a scenario like market model]

Design and develop System Dynamic model by
- Creating a stock and flow diagram
- Adding a plot to visualize dynamics
- Parameter Variation
- Calibration

[Use a case scenario like spread of contagious disease for the purpose]

Design and develop a discrete-event model that will simulate process by:

- Adding resources
- Creating 3D animation
- Modeling delivery

[Use a case situation like a company's manufacturing and shipping].

Design and develop time-slice simulation for a scenario like airport model to design how passengers move within a small airport that hosts two airlines, each with their own gate. Passengers arrive at the airport, check in, pass the security

checkpoint and then go to the waiting area. After boarding starts, each airline's representatives check their passengers' tickets before they allow them to board. Verify and validate a model developed like bank model or manufacturing model. Create defense model to stimulate aircraft behaviour. Stimulate the travelling sales man problem to compute the shortest path. Stimulate the Urban dynamics to address the scenarios like: (a) The problem of public transport line (b) To compute the time taken for train to enter the station
Specialization: PS-SCS-402: Deep Learning 8 – 10 Practicals based on the syllabus topics. OR **Specialization: PS-SCS-403: Cloud Computing – Google Cloud Platform** 8 – 10 Practicals based on the syllabus topics. OR **Specialization: PS-SCS-404: Big Data Analytics: Tools and Frameworks** 8 – 10 Practicals based on the syllabus topics. OR **Specialization: PS-SCS-405: User Experience Design** 8 – 10 Practicals based on the syllabus topics.

Scheme of Examination for Theory Courses

There will be internal and external examination for the theory courses. The weightage of internal/external and scheme of examination will be as per common guidelines provided by the University for the PG courses in the faculty of Science.

Scheme of Examination for Practical Courses

There will not be any internal examination for practical courses.

External Examination for practical courses:

The evaluation of the external examination of practical course is given below:

Sr. No.	Semester	Course Code	Particular	No of Questions	Marks Per Question	Total Marks
1	III	PS-SCS-3P1	Laboratory experiment question	2	40	80
			Journal	-	10	10
			Viva	-	10	10
		Marks for each course		100		
2	III	PS-SCS-3P2	Laboratory experiment question	2	25	50
			Journal	-	10	10
			Viva	-	10	10
			Viva on Project Proposal	Documentation	10	30
				Presentation	10	
				Viva	10	
		Total Marks		100		

Sr. No.	Semester	Course Code	Particular		No of Questions	Marks Per Question	Total Marks
1	IV	PS-SCS-4P1	Laboratory experiment question		2	40	80
			Journal		-	10	10
			Viva		-	10	10
		Total Marks				100	
2	IV	PS-SCS-4P2	Internship	Internship Conduct	Quality and Relevance	40	100
					Documentation	30	
					Presentation	30	
				Internship Viva		50	50
		Total Marks				150	
2	IV	PS-SCS-4P3	Project Dissertation	Project Conduct	Quality and Relevance	40	100
					Documentation	30	
					Presentation	30	
				Project Viva		50	50
		Total Marks				150	

Guide lines for maintenance of journals:

A student should maintain a journal with at least six practical experiments for each part of the practical course. Certified journals need to be submitted at the time of the practical examination.

Guidelines for Project Proposal in Semester - III

- Student should take a topic related to the specialization he or she is planning to take in Semester-IV.
- Should have studied the related topics in the elective he or she has chosen in Semester-II and Semester- III

- A student is expected to devote at least 2 to 3 months of study as part of topic selection and its documentation for project proposal.
- The student should be comfortable to implement the project proposal in the Semester –IV.

Guidelines for Documentation of Project Proposal in Semester –III

Student is expected to make a project proposal documentation which should contain the following:

- Title: A suitable title giving the idea about what work is proposed.
- Introduction: An introduction to the topic of around 3-5 pages, giving proper back ground of the topic discussed.
- Related works: A detailed survey of the relevant works done by others in the domain. Student is expected to refer at least 5 research papers in addition to text books and web-links in the relevant topic. It may be around 7 to 10 pages.
- Objective: A detailed objective of the proposal is needed. It may be of 1 to 2 pages.
- Methodology: A proper and detailed procedure of how to solve the problem discussed. It shall contain the techniques, tools, software and data to be used. It shall be of around 3 to 5 pages.

The report may be of around 20 pages, which needs to be signed by the teacher in charge and head of the Department. Students should submit the signed project proposal documentation at the time of viva as part of the University examination.

Guidelines for internship in Semester – IV

- Internship should be of 2 to 3 months with 8 to 12 weeks duration.
- A student is expected to find internship by himself or herself. However, the institution should assist their students in getting internship in good organizations.
- The home institution cannot be taken as the place of internship.
- A student is expected to devote at least 300 hours physically at the organization.
- Internship can be on any topic covered in the syllabus mentioned in the syllabus, not restricted to the specialization.
- Internship can be done, in one of the following, but not restricted to, types of organizations:
    - Software development firms
    - Hardware/ manufacturing firms
    - Any small scale industries, service providers like banks
    - Clinics/ NGOs/professional institutions like that of CA, Advocate etc
    - Civic Depts like Ward office/post office/police station/ punchayat.
    - Research Centres/ University Depts/ College as research Assistant for research projects or similar capacities.

Guidelines for making Internship Report in Semester –IV

A student is expected to make a report based on the internship he or she has done in an organization. It should contain the following:

- **Certificate:** A certificate in the prescribed Performa (given in appendix 1) from the organization where the internship done.
- **Evaluation form:** The form filled by the supervisor or to whom the intern was reporting, in the prescribed Performa (given in appendix 2).
- **Title:** A suitable title giving the idea about what work the student has performed during the internship.
- **Description of the organization:** A small description of 1 to 2 pages on the organization where the student has interned
- **Description about the activities done by the section where the intern has worked:** A description of 2 to 4 pages about the section or cell of the organization where the intern actually worked. This should give an idea about the type of activity a new employee is expected to do in that section of the organization.
- **Description of work allotted and actually done by the intern:** A detailed description of the work allotted and actual work performed by the intern during the internship period. Intern may give a weekly report of the work by him or her if needed. It shall be of around 7 to 10 pages.
- **Self-assessment:** A self-assessment by the intern on what he or she has learnt during the internship period. It shall contain both technical as well as inter personal skills learned in the process. It shall be of around 2 to 3 pages.

The internship report may be around 15 pages and this needs to be submitted to the external examiner at the time of University examination.

Guidelines for Research Implementation in Semester – IV

- Student should continue with topic proposed and evaluated at the Semester – III.
- The topic has to be related with the specialization he or she has chosen in the Semester – IV.
- A student is expected to devote at least 3 to 4 months of efforts for the implementation.
- Student should submit a detailed project implementation report at the time of viva.

Guidelines for Documentation of Project Implementation Report in Semester IV

A Student should submit project implementation report with following details:

- Title: Title of the project (Same as the one proposed and evaluated at the Semester II examination).
- Implementation details: A description of how the project has been implemented. It shall be of 2 to 4 pages.

- Experimental set up and results: A detailed explanation on how experiments were conducted, what software used and the results obtained. Details like screen shots, tables and graphs can come here. It shall be of 6 to 10 pages.
- Analysis of the results: A description on what the results means and how they have been arrived at. Different performing measures or statistical tools used etc. may be part of this. It shall be of 4 to 6 pages.
- Conclusion: A conclusion of the project performed in terms of its outcome (May be half a page).
- Future enhancement: A small description on what enhancement can be done when more time and resources are available (May be half a page).
- Program code: The program code may be given as appendix.

The report may be of around 20 pages (excluding program code), which needs to be signed by the teacher in charge and head of the Department. Student should submit the signed project implementation report along with evaluated copy of the project proposal documentation (of Semester –III) at the time of Project evaluation and viva as part of the University examination.

Appendix 1

(Proforma for the certificate for internship in official letter head)

This is to certify that Mr./Ms._____of _____College/Institution worked as an intern as part of her MSc course in Computer Science of University of Mumbai. The particulars of internship are given below:

Internship starting date: _____

Internship ending date: _____

Actual number of days worked: _____

Tentative number of hours worked:_____Hours

Broad area of work: _____

A small description of work done by the intern during the period:

_____
_____
_____
_____
_____
_____

Signature:

Name:

Designation:

Contact number:

Email:

(Seal of the organization)

Appendix 2

(Proforma for the Evaluation of the intern by the supervisor/to whom the intern was reporting in the organization)

Professional Evaluation of intern

Name of Intern: _____

College/institution:_____

[Note: Give a score in the 1-5 scale by putting √ in the respective cells]

Sr. No.	Particular	Excellent	Very Good	Good	Moderate	Satisfactory
1	Attendance					
2	Punctuality					
3	Adaptability					
4	Ability to shoulder responsibility					
5	Ability to work in a team					
6	Written and Oral Communication Skills					
7	Problem Solving Skills					
8	Ability to grasp new concepts					
9	Ability to complete task					
10	Quality of work done					

Comments:

_____

_____

_____

Signature:

Name:

Designation:

Contact number:

Email:

(Seal of the organization)