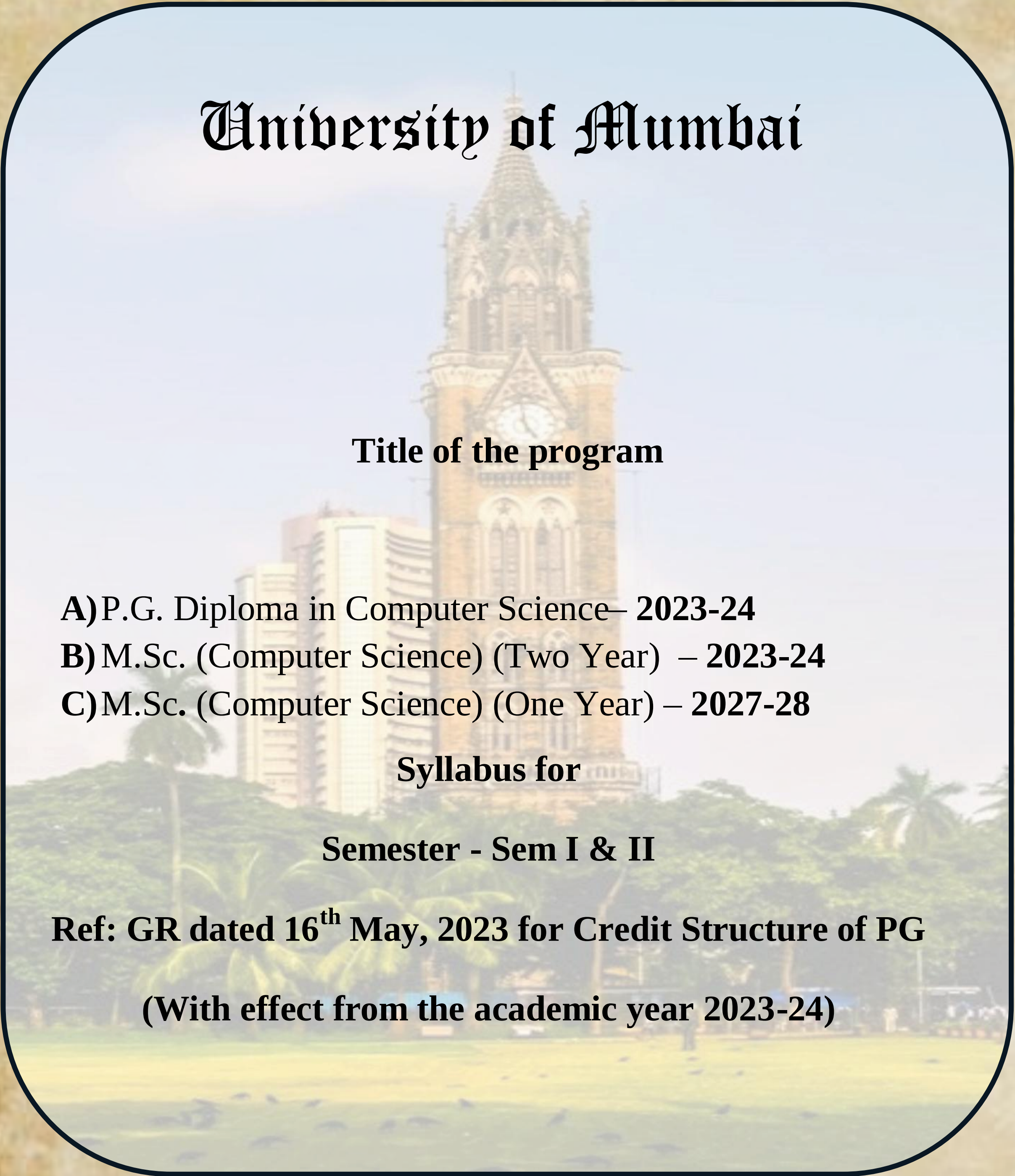
AC– 07/07/2023
Item No. –6.27(N)

As Per NEP 2020



University of Mumbai

Title of the program

A) P.G. Diploma in Computer Science– **2023-24**
B) M.Sc. (Computer Science) (Two Year) – **2023-24**
C) M.Sc. (Computer Science) (One Year) – **2027-28**

Syllabus for

Semester - Sem I & II

Ref: GR dated 16th May, 2023 for Credit Structure of PG

(With effect from the academic year 2023-24)

University of Mumbai

(As per NEP 2020)

Sr. No.	Heading	Particulars		
1	**Title of program:** O.SP-37A	**A**	P.G. Diploma in Computer Science	
	O. SP-37B	**B**	M.Sc. (Computer Science)	(Two Year)
	O. SP-37C	**C**	M.Sc. (Computer Science)	(One Year)
2	**Eligibility** O.SP-38A	**A**	B.Sc. (CS), B.Sc. (IT), B.Sc. (Data Science), B.E, B.Sc. (Physics), B.Sc. (Mathematics), B.Sc. (Statistics)	
	O.SP-38B	**B**	B.Sc. (CS), B.Sc. (IT), B.Sc. (Data Science), B.E, B.Sc. (Physics), B.Sc. (Mathematics), B.Sc. (Statistics)	
	O. SP-38C	**C**	Graduate with 4 year UG Degree (Honours/Honours with Research) with specialization in concerned subject or equivalent academic level 6.0 OR Graduate with 4 years UG Degree program with maximum credits required for award of Minor degree is allowed to take up the Post Graduate program in Minor subject provided the student has acquired the required number of credits as prescribed by the concerned Board of Studies.	
3	**Duration of program** R.SP-91	**A**	1 Year	
		B	2 Years	
		C	1 Year	
4	**Intake Capacity** R.SP-92	120		
5	**Scheme of Examination** R. SP-93	NEP 50% Internal & 50% External, Semester End Examination Individual Passing in Internal and External Examination.		

6	**Standard of Passing** R. SP-94	40%	
7	**Credit Structure:** R. SP-95A R. SP-95B	Attached herewith	
8	Semesters	**A**	Sem - I & II
		B	Sem - I, II, III & IV
		C	Sem - I & II
9	Program Academic Level	**A**	6.0
		B	6.5
		C	6.5
10	Pattern	**Semester**	
11	Status	**New**	
12	To be implemented from Academic Year Progressively	**A**	2023-24
		B	
		C	2027-28

DrJyotshnaDongardive
Head, Department of ComputerScience University of Mumbai

DrShivramGarje
Dean, Science and Technology
University of Mumbai

PREAMBLE

1. Introduction

The Master of Science in Computer Science (M.Sc. Computer Science) is an advancedprogram that combines academic research and industry standards, addressing the evolvingneeds of both the industry and research domains. The curriculum focuses on cutting-edgetechnologies and industry insights, ensuring students gain the necessary expertise to thrive inthe current landscape.

Throughout this program, students will delve into a wide range of relevant subjects. In first year,they study subjects including Machine Learning, Image Processing, Networking, Blockchain,Cloud Computing, Big Data, Computer and Network Security, Web Data Mining, and Simulationand Modelling. In the M.Sc. second year program, students will engage with major mandatorysubjects such as Web3 Technologies, Cyber Security and Risk Assessment, Ethical &Responsible AI, Deep Learning, and Big Data Analytics. They can also choose from majorelective subjects like Social Network Analysis or Data Visualization or Fuzzy Systems, as wellas Trends in Cloud Computing or Remote Sensing or Server Virtualization. Research projects inboth semesters provide practical experience and foster critical skills. This comprehensivecurriculum equips students with the latest knowledge and prepares them for diverseopportunities in computer science. These courses are carefully designed to equip students withthe skills required to tackle the challenges and opportunities in the rapidly expanding field ofComputer Science.The program is structured as a fusion of Major Mandatory and Major Elective courses,encompassing the latest trends and advancements in Computer Science. In each semester,students have the opportunity to choose one elective subject aligned with their interests from aselection of three options. The Major Mandatory courses establish a strong foundation infundamental concepts of Computer Science and Research, while the electives enhance theirknowledge for real-world applications. Practical implementation is facilitated through the use ofindustry-standard tools and simulators, such as Cisco for networking and Python forprogramming.

To further enhance the students; readiness for industry, the curriculum incorporates amandatory On Job Training (OJT) component in Semester II. This intensive training, equivalentto a full course, provides invaluable exposure to real-world scenarios within IT or IT-relatedorganizations. By applying their theoretical knowledge in practical settings, students gain first-hand experience and develop the necessary skills to thrive in the professional world.In addition to technical skills, this program also focuses on cultivating research ethics andpromoting a research-oriented mindset among learners. The inclusion of a ResearchMethodology Course helps students develop a strong research attitude, enabling them tocontribute meaningfully to the advancement of Computer Science.The comprehensive education provided by the M.Sc. in Computer Science program equipsstudents with the confidence to adapt and excel in an ever-evolving industry and academiclandscape. The curriculum continuous refinement has been made possible through thevaluable inputs, suggestions, and observations of colleagues at the University of Mumbai,experts from premier institutions, and industry professionals. We extend our gratitude to allthose who have directly or indirectly contributed to the development of this program. With thesecombined efforts, the M.Sc. in Computer Science program aims to empower students with theskills and knowledge necessary to thrive in the digital world.

2. Aims and Objectives

The aims and objectives of the M.Sc. Computer Science program collectively aim to develop well-rounded computer science professionals who are not only technically competent but also capable of contributing to research, innovation, and the overall advancement of the field.

Objectives:

- **Provide In-depth Knowledge:** The program aims to provide students with a comprehensive understanding of the key concepts, theories, and methodologies in Computer Science. It covers a range of topics including machine learning, data mining, data visualization, and data management, enabling students to develop a deep knowledge base in these areas.
- **Develop Programming Skills:** The program aims to equip students with strong programming skills by providing hands-on experience with different tools and technologies. Students will gain proficiency in designing front-end and back-end solutions, enhancing their ability to develop robust and scalable applications.
- **Foster Problem-solving Abilities:** The program aims to enhance students' problem-solving abilities by training them to approach real-world data challenges critically and creatively. Students will learn to identify problems, design appropriate data analysis strategies, and develop innovative solutions using their knowledge of Computer Science.
- **Encourage Collaboration and Teamwork:** The program aims to foster collaboration and teamwork skills among students, recognizing that computer science projects often require interdisciplinary collaboration. Students will learn to effectively communicate, collaborate, and contribute as part of a team, preparing them for collaborative work environments.
- **Foster Industry Relevance:** The program aims to stay up-to-date with industry trends and technologies to ensure graduates are well-prepared for the demands of the Computer Science job market. Through industry partnerships and internships, students will have the opportunity to gain practical experience and stay in touch with the latest advancements in the field.
- **Professional Development:** The program aims to prepare students for successful careers in the field of Computer Science. In addition to technical skills, students will develop professional skills such as teamwork, project management, and leadership. The program may also provide networking opportunities, internships, or

collaborations with industry partners to enhance students' industry readiness and employability.

- **Cultivate Research Skills**: The program aims to cultivate research skills among students by providing a Research Methodology Course and encouraging participation in research projects. Students will learn to conduct literature reviews, design experiments, analyze data, and present their findings, fostering a research-oriented mindset and contributing to the advancement of Computer Science.

By incorporating these objectives, the M.Sc. Computer Science program aims to produce graduates who possess a strong foundation in Computer Science, are adept at problem-solving and collaboration, have industry-relevant skills, and are well-prepared for both research and professional roles in the field.

3. Learning Outcomes

- Develop a solid foundation in fundamental concepts, theories, and methodologies of Computer Science.
- Offer opportunities for specialization in a chosen area of Computer Science.
- Foster a research-oriented mindset and contribute to the advancement of Computer Science.
- Prepare learners for lifelong learning, adapting to emerging technologies and industry requirements.
- Inculcate professional attitudes, leadership qualities, and social responsibility.
- Equip students with industry-relevant skills and experiences for successful careers.
- Enhance critical thinking and innovative problem-solving abilities.

4. Credit Structure of the Program (Sem I, II, III & IV)

R. SP-95A

Year	**Level**	**Sem**	**Major**				**RM**	**OJT/FP**	**RP**	**Cum.Cr.**	**Degree**
			Mandatory			**Electives**					
1	6.0	SemI	2*4+2*2+2			4	4	-	-	22	PG Diploma(after 3YearsDegree)
			Applied Signal &Image Processing (501)	TH	4	NoSQL Technologies (506a)2 TH+2PR **(OR)** Robotic(506b) 2 TH +2PR **(OR)** UI/UX Design (506c) 2 TH +2PR	Research Methodology (510)				
			Applied Signal &Image Processing Practical (502)	PR	2						
			Software Defined Networking(503)	TH	4						
			Software Defined Networking Practical(504)	PR	2						
			Principles of Compiler Design(505)	TH	2						
		SemII	2*4+2*2+2			4	-	OJT(517) 4	-	22	
			Machine Learning (511)	TH	4	Bioinformatics(516a) 2 TH +2PR **(OR)** Embedded and IoT Technology (516b) 2 TH +2PR **(OR)** Web Data Analytics(516c) 2 TH +2PR					
			Machine Learning Practical(512)	PR	2						
			Natural Language Processing(513)	TH	4						
			Natural Language Processing Practical(514)	PR	2						
			Simulation and Modelling (515)	TH	2						
Cum.Cr. For PG Diploma			28			8	4	4		44	
Exit Option:PG Diploma(44credits)after Three Year UG Degree											

R. SP-95B

Year	Level	Sem (2yr)	Major				RM	OJT/FP	RP	Cum.Cr.	Degree
			Mandatory			Electives					
2	6.5	Sem III	2*4+2*2+2			4	-	-	RP (607) 4	22	PG Degree after 3-yr UG
			Web3Technologies (601)	TH	4	Social Network Analysis (606a) 2 TH +2PR **(OR)** Data Visualization (606b) 2 TH+2 PR **(OR)** Fuzzy Systems (606c) 2 TH +2PR					
			Web3 Technologies Practical (602)	PR	2						
			Cyber Security and Risk Assessment (603)	TH	4						
			Cyber Security and Risk Assessment Practical (604)	PR	2						
			Ethical & Responsible AI (605)	TH	2						
		Sem IV	2*4+2*2			4	-	-	RP (616)	22	
			Deep Learning(611)	TH	4	Trends in cloud computing (615a)2 TH +2PR **(OR)** Remote Sensing (615b)2 TH +2PR **(OR)** Server Virtualization (615c)2 TH +2PR					
			Deep Learning Practical (612)	PR	2						
			Big Data Analytics (613)	TH	4						
			Big Data Analytics Practical (614)	PR	2						
Cum.Cr. For 1 Yr PG Degree			26			8			10	44	
Cum.Cr. For 2 Yr PG Degree			54			16	4	4	10	88	

Garje

DrJyotshnaDongardive
Head, Department of Computer Science University of Mumbai

DrShivramGarje
Dean, Science and Technology
University of Mumbai

Sem-I

SEMESTER- I

Programme Name: M.Sc. Computer Science (Semester I)	**Course Name:** Applied Signal and Image Processing
Total Credits: 04	**Total Marks:** 100
College assessment: 50	**University assessment:** 50

Prerequisite: Fundamental knowledge of mathematics, digital signal processing, programming, and image processing.

Course Outcome:

- Understand and apply the fundamentals of digital signal processing and frequency domain operations for image analysis.
- Gain proficiency in image processing techniques such as intensity transformations, histogram processing, and smoothing.
- Develop skills in edge detection and image segmentation using various algorithms and approaches.
- Utilize morphological operations for image enhancement, feature extraction, and noise reduction.
- Apply advanced image processing techniques including feature detection, descriptors, and segmentation algorithms for complex image analysis and understanding.

<table>
<tr><th>Course Code</th><th>Course Title</th><th>Total Credits</th></tr>
<tr><td><b>PSCS501</b></td><td><b>Applied Signal and Image Processing</b></td><td><b>04</b></td></tr>
<tr><td colspan="2"><b>MODULE - I</b>
<b>Unit 1: Fundamentals of Digital Signals Processing</b>
Periodic signals, Spectral decomposition, Signals, Reading and writing Waves, Spectrums, Wave objects, Signal objects ,Noise: Uncorrelated noise, Integrated spectrum, Brownian noise, Pink Noise, Gaussian noise; Autocorrelation: Correlation, Serial correlation, Autocorrelation, Autocorrelation of periodic signals, Correlation as a dot product Frequency domain Operations: Representing Image as Signals, Sampling and Fourier Transforms, Discrete Fourier Transform, Convolution and Frequency Domain Filtering, Smoothing using lowpass filters, Sharpening using high-pass filters. Fast Fourier Transforms.

<b>Unit 2:Image Processing fundamentals and Pixel Transformation</b>
Definition, Application of Image Processing, Image Processing Pipeline, Tools and Libraries for Image Processing, Image types and files formats. Intensity Transformations- Log Transform, Power-law Transform, Contrast Stretching, Thresholding Histogram Processing- Histogram Equalization and Histogram Matching; Linear and Non-linear smoothing of Images, Sharpening of images Image Derivative: Derivatives and gradients, Laplacian, the effect of noise on gradient computation</td><td><b>02</b></td></tr>
</table>

MODULE - II **Unit 3:Structural and Morphological Operations** Edge Detection: Sobel, Canny Prewitt, Robert edge detection techniques, LoG and DoG filters, Image Pyramids: Gaussian Pyramid, Laplacian Pyramid Morphological Image Processing: Erosion, Dilation, Opening and closing, Hit-or-Miss Transformation, Skeletonizing, Computing the convex hull, removing small objects, White and black top-hats, Extracting the boundary, Grayscale operations **Unit 4: Advanced Image Processing Operations** Extracting Image Features and Descriptors: Feature detector versus descriptors, Boundary Processing and feature descriptor, Principal Components, Harris Corner Detector, Blob detector, Histogram of Oriented Gradients, Scale-invariant feature transforms, Haar-like features Image Segmentation: Hough Transform for detecting lines and circles, Thresholding and Otsu’s segmentation, Edge-based/regionbased segmentation Region growing, Region splitting and Merging, Watershed algorithm, Active Contours, morphological snakes, and GrabCut algorithms	**02**

Text Books:

1. Digital Image Processing by Rafael Gonzalez & Richard Woods, Pearson; 4th edition, 2018.
2. Think DSP: Digital Signal Processing in Python by Allen Downey, O'Reilly Media; 1st edition (August 16, 2016).

Reference Books:

1. Understanding Digital Image Processing, VipinTyagi, CRC Press, 2018.
2. Digital Signal and Image Processing by Tamal Bose, John Wiley 2010.
3. Hands-On Image Processing with Python by SandipanDey,Packt Publishing, 2018.
4. Fundamentals of Digital Images Processing by A K Jain, Pearson, 2010

Programme Name: M.Sc. Computer Science (Semester I) **Total Credits:** 02	**Course Name:** Applied Signal and Image Processing Practical **Total Marks:** 50 **University assessment:** 50

Prerequisite: Knowledge of Python Programming, concepts of signal and image processing

Course Outcome:

- Apply signal processing techniques: Demonstrate upsampling, downsampling, and FFT for analyzing signals.
- Signal analysis and correlation: Create triangle signals, compute correlations between segments, and plot signal segments.
- Sound and image processing: Implement convolution operations, template matching, and image derivatives for analysis.
- Intensity transformations and enhancement: Perform log, power-law, contrast adjustments, histogram equalization, and thresholding on images.
- Image filtering and feature extraction: Apply gradient, Laplacian, and noise smoothing techniques to enhance image features.
- Edge detection and segmentation: Utilize Sobel, Canny, and morphological operations for edge detection and region-based segmentation.

Course Code	Course Title	Credits
PSCSP502	**Applied Signal and Image Processing Practical**	**02**
Note: - The following set of practical should be implemented in Scrape, python: **Link: -Python:https://www.python.org/downloads/**		
1	Write program to demonstrate the following aspects of signal processing on suitable data 1. Upsampling and downsampling on Image/speech signal 2. Fast Fourier Transform to compute DFT	
2	Write program to perform the following on signal 1. Create a triangle signal and plot a 3-period segment. 2. For a given signal, plot the segment and compute the correlation between them.	
3	Write program to demonstrate the following aspects of signal on sound/image data 1. Convolution operation 2. Template Matching	
4	Write program to implement point/pixel intensity transformations such as 1. Log and Power-law transformations 2. Contrast adjustments 3. Histogram equalization 4. Thresholding, and halftoning operations	
5	Write a program to apply various enhancements on images using image derivatives by	

	implementing Gradient and Laplacian operations.
6	Write a program to implement linear and nonlinear noise smoothing on suitable image or sound signal.
7	Write a program to apply various image enhancement using image derivatives by implementing smoothing, sharpening, and unsharp masking filters for generating suitable images for specific application requirements
8	Write a program to Apply edge detection techniques such as Sobel and Canny to extract meaningful information from the given image samples
9	Write the program to implement various morphological image processing techniques.
10	Write the program to extract image features by implementing methods like corner and blob detectors, HoG and Haar features
11	Write the program to apply segmentation for detecting lines, circles, and other shapes/ objects. Also, implement edge-based and region-based segmentation.

Programme Name: M.Sc. Computer Science (Semester I)	**Course Name:** Software Defined Networking
Total Credits: 04	**Total Marks:** 100
College assessment: 50	**University assessment:** 50

Prerequisite: Basic Networking concepts.

Course Outcome:

- Understand computer networking concepts, OSI/TCP-IP models, and routing protocols.
- Gain knowledge and skills in Software Defined Networking (SDN) architecture, OpenFlow, and application development.
- Comprehend Network Functions Virtualization (NFV), cloud computing, and IoT integration in modern network architectures.
- Design and implement switching techniques, routing protocols, multicast, MPLS, traffic filtering, and routing redistribution.
- Develop network design and deployment skills for efficient and secure routing, traffic management, and integration of network components.

<table>
<tr><th>Course Code</th><th>Course Title</th><th>Total Credits</th></tr>
<tr><td><b>PSCS503</b></td><td><b>Software Defined Networking</b></td><td><b>04</b></td></tr>
<tr><td colspan="2"><b>MODULE - I</b>
<b>Unit 1: Introduction to Computer Networking</b>
Basic Concepts and Definitions: LAN, MAN, WAN, AD-Hoc, Wireless Network, Understanding the layered architecture of OSI/RM and TCP-IPModel, Concepts and implementation of IPV4 and IPV6, Study of various network Routing protocols, Introduction to Transport layer and Application layer protocols.

<b>Unit 2: Software Defined Networking</b>
Elements of Modern Networking, Requirements and Technology, SDN: Background and Motivation, SDN Data Plane and OpenFlow, SDN
Control Plane, SDN Application Plane</td><td><b>02</b></td></tr>
<tr><td colspan="2"><b>MODULE - II</b>
<b>Unit 3: Network Functions Virtualization</b>
Concepts and Architecture, NFV Functionality, Network Virtualization Quality of Service, Modern Network Architecture: Clouds and Fog, Cloud Computing, The Internet of Things: Components

<b>Unit 4: Design and implementation of Network</b>
Understand and implement Layer 2/3 switching techniques (VLAN /TRUNKING/ Managing Spanning Tree), Implementation of OSPF V2 and V3, Implementation BGP, Implementation Multicast Routing, Implementation of MPLS, Implementation of Traffic</td><td><b>02</b></td></tr>
</table>

Filtering by using Standard and Extended Access Control List, Implementation of Routing redistribution, Implementation	

Text Books:

1. TCPIP Protocol Suite, Behrouz A Forouzan , McGraw Hill Education; 4th edition, Fourth Edition, 2017
2. Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud, William Stallings, Addison-Wesley Professional, 2016.
3. Software Defined Networks: A Comprehensive Approach, Paul Goransson and Chuck Black, Morgan Kaufmann Publications, 2014
4. SDN - Software Defined Networks by Thomas D. Nadeau & Ken Gray, O'Reilly, 2013

Programme Name: M.Sc. Computer Science (Semester I) **Total Credits:** 02	**Course Name:** Software Defined Networking Practical **Total Marks:** 50 **University assessment:** 50

Prerequisite: Basic Networking concepts, Knowledge of Cisco Packet Tracer.

Course Outcome:

- Implement various network protocols and technologies, including IP SLA, IPv4 ACLs, SPAN, SNMP, and NetFlow.
- Configure network connectivity and address translation using GRE tunnels, VTP, NAT, and inter-VLAN routing.
- Understand and optimize network spanning tree operation through STP topology changes, RSTP, and advanced STP mechanisms.
- Establish and manage advanced networking features such as EtherChannel, OSPF, BGP, and IPsec VPNs.
- Simulate and analyze Software-Defined Networking (SDN) environments using OpenDaylight and Mininet/OpenFlow.

Course Code	Course Title	Credits
PSCSP504	**Software Defined Networking Practical**	**02**
Note: All the Practical’s should be implemented using GNS3/EVENG/CISCO VIRL Link: GNS3:https://www.gns3.com/software/download EVE-NG: https://www.eve-ng.net/index.php/download/CISCO VIRL: https://learningnetwork.cisco.com/s/question/0D53i00000Kswpr/virl15-download		
1	Implement IP SLA (IP Service Level Agreement)	
2	Implement IPv4 ACLs a) Standard ACL b) Extended ACL	
3	a) Implement SPAN Technologies (Switch Port Analyzer) b) Implement SNMP and Syslog c) Implement Flexible NetFlow	
4	a) Implement a GRE Tunnel b) Implement VTP c) Implement NAT	
5	Implement Inter-VLAN Routing	
6	Observe STP Topology Changes and Implement RSTP	

	a) Implement Advanced STP Modifications and Mechanisms b) Implement MST
7	a) Implement EtherChannel b) Tune and Optimize EtherChannel Operations
8	OSPF Implementation a) Implement Single-Area OSPFv2 b) Implement Multi-Area OSPFv2 c) OSPFv2 Route Summarization and Filtering d) Implement Multiarea OSPFv3
9	a) Implement BGP Communities b) Implement MP-BGP c) Implement eBGP for IPv4 d) Implement BGP Path Manipulation
10	a) Implement IPsec Site-to-Site VPNs b) Implement GRE over IPsec Site-to-Site VPNs c) Implement VRF Lite
11	Simulating SDN with a) OpenDaylight SDN Controller with the Mininet Network Emulator b) OFNet SDN network emulator
12	Simulating OpenFlow Using MININET

Programme Name: M.Sc. Computer Science (Semester I) **Total Credits:** 02 **College assessment:** 25	**Course Name:** Principles of Compiler Design **Total Marks:** 50 **University assessment:** 25

Prerequisite: Programming Language concepts, Data Structures and Algorithms, Discrete Mathematics.

Course outcomes:

- Understand the theoretical foundations and concepts underlying the design and implementation of compilers.
- Acquire knowledge about the different phases of the compilation process
- Learn how to design and implement lexical analyzers and parsers
- Gain hands-on experience in building semantic analyzers
- Understand intermediate code generation and Implement optimization techniques
- Gain practical experience in code generation
- Familiarity with runtime environments and Develop skills in error handling and debugging
- Explore advanced topics in compiler design and Apply knowledge to practical projects

Course Code	Course Title	Total Credits
PSCS505	**Principles of Compiler Design**	**02**
MODULE - I **Unit 1: Front end of Compiler** **Introduction to Compiler Design**: Role and importance of compilers, Phases of compilation process, Compiler architecture and components **Lexical Analysis**: Role of lexical analyzer, Regular expressions and finite automata, Lexical analyzer generators (e.g., Lex) **Syntax Analysis**: Role of parser, Context-free grammars, Top-down parsing (LL parsing) Bottom-up parsing (LR parsing), Syntax analyzer generators (e.g., Yacc/Bison) **Semantic Analysis**: Role of semantic analyzer, Symbol table management, Type checking and type systems, Attribute grammars **Intermediate Code Generation**: Intermediate representations (IR), Three-address code generation, Quadruples and triples, Syntax-directed translation **Unit 2: Back end of Compiler** **Code Optimization**: Data flow analysis, Common subexpression elimination, Constant folding and propagation, Loop optimization techniques **Code Generation**: Code generation techniques, Target machine description, Register allocation, Instruction selection and scheduling		**02**

Runtime Environments: Activation records and stack management. Heap memory management, Call and return mechanisms, Exception handling Lexical and Syntax Error Handling: Error recovery strategies Error reporting and handling **Introduction to Compiler Tools, Techniques and Advanced Topics in Compiler Design**: Lexical and syntax analyzer generators, Code generation frameworks (e.g., LLVM), Debugging and testing compilers, Just-in-time (JIT) compilation, Parallel and concurrent programming support, Compiler optimization frameworks, Domain-specific language (DSL) compilation	

Text Books:

1. Compilers: Principles, Techniques, and Tools" by Alfred V. Aho, Monica S. Lam, Ravi Sethi, and Jeffrey D. Ullman 2nd Edition, Pearson Publication, 2006 ISBN-13: 978-0321486813

Reference Books:

1. Modern Compiler Implementation in C" by Andrew W. Appel, 3rd Edition, Cambridge University Press, 2020, ISBN-13: 978-1108426631
2. Principles of Compiler Design" by D. M. Dhamdhere, 2nd Edition Publisher: McGraw-Hill Education, 2017, ISBN-13: 978-9339204608

ELECTIVES

Programme Name: M.Sc. Computer Science (Semester I) **Total Credits:** 02 **College assessment:** 25	**Course Name:** NoSQL Technologies **Total Marks:** 50 **University assessment:** 25

Prerequisite: Basic understanding of databases, SQL concepts, and familiarity with programming languages like Java or Python.

Course Outcome:

Upon the successful completion of this course, students will be able to:

- Understand NoSQL characteristics, storage types, and advantages/drawbacks.
- Interface and interact with MongoDB, Redis, HBase, and Apache Cassandra effectively.
- Comprehend storage architecture in NoSQL, including column-oriented, document stores, and key/value stores.
- Perform CRUD operations proficiently, including data creation, access, update, and deletion.
- Query NoSQL stores using MongoDB features, accessing HBase data, and querying Redis.
- Apply indexing and ordering concepts in NoSQL databases like MongoDB, CouchDB, and Cassandra.
- Manage transactions and ensure data integrity in NoSQL, including distributed ACID systems.
- Utilize NoSQL effectively in the cloud, such as Google App Engine Data Store and Amazon SimpleDB.

Course Code	Course Title	Total Credits
PSCS506a	**NoSQL Technologies**	**02**
MODULE - I **Unit 1: Introduction to NoSQL and Interfacing with NoSQL Data Stores** **Basics Introduction to NoSQL:** Characteristics of NoSQL, NoSQL Storage types, Advantages and Drawbacks, NoSQL Products Interfacing and interacting with NoSQL: Storing Data in and Accessing Data from MongoDB, Redis, HBase and Apache Cassandra, Language Bindings for NoSQL Data Stores **Understanding the storage architecture:** Working with ColumnOriented Databases, HBase Distributed Storage Architecture, Document Store Internals, Understanding Key/Value Stores in Memcached and Redis, Eventually Consistent Non-relational **Databases Performing CRUD operations:** Creating Records, Accessing Data, Updating and Deleting Data **Unit 2: Querying, Indexing, and Data Management in NoSQL Databases**		**02**

Querying NoSQL Stores: Similarities Between SQL and MongoDB Query Features, Accessing Data from Column-Oriented Databases Like HBase, Querying **Redis Data** Stores Indexing and Ordering Data Sets: Essential Concepts Behind a Database Index, Indexing and Ordering in MongoDB, ouchDB and Apache Cassandra **Managing Transactions and Data Integrity:** RDBMS and ACID, Distributed ACID Systems, Upholding CAP, Consistency Implementations Using NoSQL in The Cloud: Google App Engine Data Store, Amazon SimpleDB	

TextBooks:

1. QL &NoSQL Databases, Andreas Meier · Michael Kaufmann, Springer Vieweg, 2019
2. Professional NoSQLbyShashank Tiwari, Wrox-John Wiley & Sons, Inc, 2011
3. SQL &NoSQL Databases, Andreas Meier · Michael Kaufmann, Springer Vieweg, 2019
4. NoSQL: Database for Storage and Retrieval of Data in Cloud, Ganesh Chandra Deka, CRC Press, 2017
5. Demystifying NoSQL by Seema Acharya, Wiley, 2020

Programme Name: M.Sc. Computer Science (Semester I) **Total Credits:** 02	**Course Name:** NoSQL Technologies Practical **Total Marks:** 50 **University assessment:** 50

Prerequisite: Basic understanding of databases, familiarity with SQL, proficiency in a programming language, and knowledge of data modeling concepts.

Course Outcome:

1. Set up and configure various NoSQL databases, including MongoDB, Redis, HBase, and Apache Cassandra.
2. Perform CRUD operations and retrieve data from different NoSQL databases using appropriate query languages and commands.
3. Understand the storage architecture and internal workings of different NoSQL databases, such as column-oriented databases, document stores, and key/value stores.
4. Implement data indexing and explore its impact on query performance in MongoDB and other NoSQL databases.
5. Utilize NoSQL technologies in real-world scenarios, such as caching with Redis, data storage in Google App Engine Data Store, and Amazon SimpleDB data management.

Course Code	Course Title	Credits
PSCSP506a	**NoSQL Technologies Practical**	**02**
1	**Lab Exercise: Setting up and Exploring MongoDB** a) Install MongoDB on your local machine or lab server. b) Create a new MongoDB database and collection. c) Insert sample data into the collection. d) Retrieve and display data from the collection using MongoDB queries.	
2	**Interacting with Redis** a) Install Redis on your lab server or local machine. b) Store and retrieve data in Redis using various data structures like strings, lists, and sets. c) Implement basic Redis commands for data manipulation and retrieval	
3	**Working with HBase** a) Set up an HBase cluster in a lab environment. b) Create an HBase table and define column families. c) Insert sample data into the table. d) Perform CRUD operations and retrieval of data in HBase.	
4	**Apache Cassandra Operations** a) Install and configure Apache Cassandra in a lab environment. b) Create a keyspace and define a table schema.	

	c) Insert data into the table. d) Perform CRUD operations and query data from Apache Cassandra.
5	**Querying MongoDB and HBase** a) Write and execute MongoDB queries to retrieve specific data from a collection. b) Perform queries on HBase tables using HBase shell commands.
6	**Redis Data Manipulation** a) Use Redis commands to manipulate and modify data stored in different data structures. b) Retrieve specific data using Redis query operations.
7	**Implementing Indexing in MongoDB** a) Create an index on a specific field in a MongoDB collection. b) Measure the impact of indexing on query performance.
8	**Data Storage in Redis** a) Implement caching functionality using Redis as a cache store. b) Store and retrieve data from Redis cache using appropriate commands.
9	**Using Google App Engine Data Store** a) Create a project in Google App Engine and set up the Data Store. b) Store and retrieve data from the Data Store using the provided API.
10	**Amazon SimpleDB Data Management** 1. Task 1: Set up an Amazon SimpleDB domain for data storage. 2. Task 2: Store and retrieve data from the SimpleDB domain using appropriate commands or APIs.

Programme Name:M.Sc. Computer Science (Semester I) **Total Credits:** 02 **College assessment:** 25	**Course Name: Robotics** **Total Marks:** 50 **University assessment:** 25

Pre requisite: Knowledge of Basic concepts of IoT.

Course Outcome:

- Leverage the features of the Raspberry Pi OS
- Discover how to configure a Raspberry Pi to build an AI-enabled robot
- Interface motors and sensors with a Raspberry Pi
- Code robot to develop engaging and intelligent robot behavior
- Explore AI behavior such as speech recognition and visual processing

Course Code	Course Title	Total Credits
PSCS506b	**Robotics**	**02**
MODULE - I **Unit 1:Introduction to Robotics** **Introduction to Robotics:** What is a robot? Examples of Advanced and impressive robots, Robots in the home, Robots in industry, Exploring Robot Building Blocks - Code and Electronics Technical requirements, Introducing the Raspberry Pi - Starting with Raspbian Technical requirements, Raspberry Pi controller on a robot **Building Robot Basics Technical requirements:** Robot chassis kit with wheels and motors, a motor controller, Powering the robot, Test fitting the robot, Assembling the base. **Robot Programming:** Programming technique, adding line sensors to our robot, creating line-sensing behaviour, and Programming RGB Strips in robot. **Unit 2:Servo Motors** **Motors:**Use and control of servo motors, pan, and tilt mechanism, Distance sensors, Introduction to distance sensors and their usage Connecting distance sensors to robot and their testing. Creating a smart object avoidance behaviour. Creating a menu to select different robot behaviours, Distance and speed measuring sensors—encoders and odometry **Robot Vision and Voice Communication**: Setting up a Raspberry Pi Camera on the robot (software and hardware), Check the robot vision on a phone or laptop, Mask images with RGB strips, Colors, masking, and filtering – chasing		**02**

colouredobjects, detecting faces with Haar cascades, Finding objects in an image, Voice Communication with a robot	

Text Books:

1. Danny Staple, Robotics Programming, Packt Publishing, 2nd edition, Feb 2021
2. Saeed B. Niku, Introduction to Robotics: Analysis, Control, Applications, Wiley, 3rd Edition, 2019

Reference Books:

1. D. K. Pratihar, Fundamentals of Robotics. Narosa Publication, 2016
2. Lentin Joseph, Learning Robotics Using Python, Packt Publishing Ltd., May 2015

Programme Name: M.Sc. Computer Science (Semester I) **Total Credits:** 02	**Course Name:** Robotics Practical **Total Marks:** 50 **University assessment:** 50

Pre requisite: Knowledge of Basic concepts of IoT.

Course Outcome:

- Leverage the features of the Raspberry Pi OS
- Discover how to configure a Raspberry Pi to build an AI-enabled robot
- Interface motors and sensors with a Raspberry Pi
- Code robot to develop engaging and intelligent robot behavior
- Explore AI behaviour such as speech recognition and visual processing

Course Code	Course Title	Credits
PSCSP506b	**Robotics Practical**	**02**
Note: Following practical can be performed using Python and simulators, Raspberry Pi, andother hardware devices.		
1	Making a Raspberry Pi headless, and reaching it from the network using WiFi and SSH	
2	Using sftp upload files from PC.	
3	Write Python code to test motors.	
4	Write a script to follow a predetermined path	
5	Develop Python code for testing the sensors.	
6	Add the sensors to the Robot object and develop the line-following behaviour code.	
7	Using the light strip develop and debug the line follower robot	
8	Add pan and tilt service to the robot object and test it	
9	Create an obstacle avoidance behavior for robot and test it.	
10	Detect faces with Haar cascades	
11	Use the robot to display its camera as a web app on a phone or desktop, and then use the camera to drive smart color and face-tracking behaviours	
12	Use a Raspberry Pi to run the Mycroft environment and connect it to a speaker/microphone combination	

Programme Name: M.Sc. Computer Science (Semester I)	**Course Name:** UI/UX Design
Total Credits: 02	**Total Marks:** 50
University assessment: 25	**College assessment:** 25

Prerequisite: Basic understanding of design principles, familiarity with digital interfaces, creativity, and proficiency in using design tools or software.

Course Outcome:

To Understand Latest UI patterns

- Understand iterative user-centered design of graphical user interfaces
- Apply the user Interfaces to different devices and requirements,
- Create high quality professional documents and artifacts related to the design process.

<table>
<tr><th>Course Code</th><th>Course Title</th><th>Total Credits</th></tr>
<tr><td><b>PSCS506c</b></td><td><b>UI/UX Design</b></td><td><b>02</b></td></tr>
<tr><td colspan="2"><b>MODULE I</b>
<b>Unit 1: Introduction to UI</b>
<b>What is User Interface Design (UI)</b>: The Relationship Between UI and UX, Roles in UI/UX, A Brief Historical Overview ofInterface Design, Interface Conventions, Approaches to Screen Based UI, Template vs Content, Formal Elements ofInterface Design, Active Elements of Interface Design, Composing the Elements of Interface Design, UI Design Process,Visual Communication design component in Interface Design

<b>Unit 2: Introduction to UX</b>
<b>UX Basics:</b> Foundation of UX design, Good and poor design, Understanding Your Users, Designing the ExperienceElements of user Experience, Visual Design Principles, Functional Layout, Interaction design, Introduction to the Interface, Navigation Design, User Testing, Developing and Releasing Your Design
<b>Design Tools:</b> Interviews, writing personas: user and device personas, User Context, Building Low Fidelity Wireframe and High-Fidelity Polished Wireframe Using wireframing Tools, Creating the working Prototype using Prototyping tools, Sharing and Exporting Design</td><td><b>02</b></td></tr>
</table>

Text Books:

1. A Project Guide to UX Design: For user experience designers in the field or in the making (2nd. ed.). Russ Unger and Carolyn Chandler. New Riders Publishing, USA, 2012.
2. The Elements of User Experience: User-Centered Design for the Web and Beyond, Second Edition Jesse James Garrett, Pearson Education. 2011.
3. The Essential Guide to User Interface Design: An Introduction to GUI Design Principles and Techniques, Third Edition Wilbert O. Galitz , Wiley Publishing, 2007.
4. The UX Book Process and Guidelines for Ensuring a Quality User Experience, Rex Hartson and Pardha S. Pyla, Elsevier, 2012

Programme Name: M.Sc. Computer Science (Semester I) **Total Credits:** 02	**Course Name:** UI/UX Design Practical **Total Marks:** 50 **University assessment:** 50

Prerequisite: Proficiency in design principles, familiarity with design software/tools, basic understanding of user-centred design principles, creativity, and knowledge of HTML/CSS for web design tasks.

Course Outcome:

- Demonstrate proficiency in applying design principles to create visually appealing and user-friendly interfaces.
- Utilize design software and tools effectively to develop high-quality design solutions.
- Apply user-centred design principles to understand and meet user needs and expectations in design projects.
- Demonstrate creative thinking and problem-solving skills in designing interfaces that engage and delight users.
- Develop competence in using HTML/CSS to implement and showcase designs in web-based environments.

Course Code	Course Title	Credits
PSCSP506c	**UI/UX Design Practical**	**02**

1	Design appropriate UX element list for Yoga Day.
2	Design for Digital-Visual design system for a brand. Take any brand of your choice
3	Design for social media Project. Develop a web page in similar manner.
4	Design for devices: understanding web & mobile. Design a simple web interface for mobile.
5	Reading user personas and empathy maps. Based on random survey identify your customer.
6	Design a simple low-fidelity wireframe.
7	Design a simple user interface with Colour & typography for UI
8	Design simple e commerce interface
9	Design a simple homepage for mobile shopee
10	Design a web interface for 2 different brands.

Programme Name: M.Sc. Computer Science (Semester I) **Total Credits:** 04 **College assessment:** 50	**Course Name:** Research Methodology **Total Marks:** 100 **University assessment:** 50

Prerequisite: Basic knowledge in the subject, Statistics and academic writing skills.

Course Outcome:

- Define research and its significance in computer science.
- Identify different types of research methodologies and select appropriate ones for specific research questions.
- Develop a research design that includes data collection methods and ethical considerations.
- Analyze research data using statistical software or programming languages.
- Interpret research findings and draw valid conclusions.
- Write and format research papers following academic conventions.
- Prepare and deliver effective research presentations.
- Navigate the peer-review process and revise research papers accordingly.
- Understand the process of submitting research to conferences and journals.

Course Code	Course Title	Total Credits
PSCS510	**Research Methodology**	**04**
MODULE – I **Unit 1: Introduction to Research Methodology** **Understanding Research:** Definition and significance of research, Types of research, Research approaches: Quantitative vs. Qualitative, The role of research in computer science. **Research Process and Ethics:** Steps in the research process, Research ethics and responsible conduct, Formulating research questions and objectives, Research problem identification. **Research Design and Sampling:** Principles of research design, Types of research design, Sampling techniques, Choosing the appropriate research design. **Unit 2: Quantitative Research Methods** **Data Collection and Measurement:** Primary and secondary data, Data collection methods: Surveys, questionnaires, and observations, Measurement scales: Nominal, ordinal, interval, and ratio. **Statistical Data Analysis:** Measures of central tendency and variability, Data visualization techniques. **Inferential Data Analysis:** Hypothesis testing, Parametric and non-parametric tests, Statistical software tools.		**02**

MODULE – II **Unit 3: Qualitative Research Methods** **Qualitative Data Collection and Analysis:** Qualitative research methods, Data collection techniques: Interviews focus groups, Coding and interpretation of qualitative data. **Content Analysis:** What is content analysis? Types of content analysis (e.g., thematic analysis, textual analysis), Conducting content analysis in computer science research, Using software for content analysis. **Mixed-Methods Research and Reporting:** Combining quantitative and qualitative methods, Reporting research findings, Ethical considerations in mixed-methods research. **Unit 4: Research Communication and Publishing** **Writing Research Papers:** Structure of a research paper, Writing the abstract, introduction, literature review, and methodology sections, Citation and referencing. **Research Presentations and Visualizations:** Preparing effective research presentations, Visual aids and slide design, Presenting quantitative and qualitative findings **Publishing Research:** Journals vs. conferences, The peer-review process, Preparing manuscripts for submission, Ethical considerations in publishing including data privacy and security in publications.	**02**

Text Books:

1. Research Methodology: Methods and Techniques, C.R. Kothari, New Age International

Sem-II

SEMESTER II

Programme Name: M.Sc. Computer Science (Semester II)	**Course Name:** Machine Learning
Total Credits: 04	**Total Marks:** 100
College assessment: 50	**University assessment:** 50

Prerequisite: Mathematics (calculus and linear algebra), programming skills (Python), statistics and probability knowledge, data analysis experience.

Course Outcome:

- Develop a solid understanding of the fundamentals of machine learning, including its types and applications.
- Apply various machine learning techniques such as linear regression, logistic regression, and decision trees to train models and make predictions.
- Gain proficiency in using support vector machines (SVM) for classification and regression tasks.
- Explore the fundamentals of deep learning and artificial neural networks, including their architecture and activation functions.
- Acquire practical skills in implementing machine learning algorithms using the TensorFlow framework and analyzing performance measures for model evaluation.

Course Code	Course Title	Total Credits
PSCS511	**Machine Learning**	**04**
MODULE I **Unit 1:The Fundamentals of Machine Learning** Understanding Machine Learning,Need and Relevance of Machine Learning, Types of Machine Learning, Supervised Learning, Unsupervised Learning & Reinforcement Learning. Challenges of Machine Learning, Testing and Validation. Classification, MNIST Dataset, Performance Measures, Confusion Matrix, Precision and Recall, Precision/Recall Tradeoff, The ROC Curve, Multiclass Classification, Error Analysis. **Unit 2: Training Models** Linear Regression, Gradient Descent, Batch Gradient Descent, Stochastic Gradient Descent, Mini-batch Gradient Descent, Polynomial Regression, Learning Curves, The Bias/Variance Tradeoff, Ridge Regression, Lasso Regression, Early Stopping, Logistic Regression, Decision Boundaries, Softmax Regression, Cross Entropy.		**02**

MODULE II **Unit 3:Support Vector Machines** Linear SVM Classification, Soft Margin Classification, Nonlinear SVM Classification, Polynomial Kernel, Gaussian RBF Kernel, SVM Regression, Decision Trees, Training and Visualizing a Decision Tree, Making Predictions, The CART Training Algorithm, Gini Impurity vs Entropy, Regularization Hyperparameters. **Unit 4: Fundamentals of Deep Learning** What is Deep Learning? Need Deep Learning? Introduction to Artificial Neural Network (ANN), Core components of neural networks, Multi-Layer Perceptron (MLP), Activation functions, Sigmoid, Rectified Linear Unit (ReLU), Introduction to Tensors and Operations, Tensorflow framework	**02**

Text Books:

1. Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow Concepts, Tools, and Techniques to Build Intelligent Systems by AurélienGéron, Second Edition, O'reilly, 2019
2. Deep Learning with Python by François Chollet Published by Manning 2018
3. Reinforcement Learning: An Introduction by Richard S. Sutton and Andrew G. Barto, Second Edition ,2014

Reference Books:

1. Introduction to Machine with Python - A Guide for Data Scientists by Andreas C. Müller & Sarah Guido O'reilly 2016
2. Artificial Neural Networks with TensorFlow 2 ANN Architecture Machine Learning Projects PoornachandraSarang by Apress, 2021

Programme Name: M.Sc. Computer Science (Semester I**I)** **Total Credits:** 02	**Course Name:** Machine Learning Practical **Total Marks:** 50 **University assessment:** 50

Prerequisite: Basic understanding of machine learning concepts, familiarity with Python programming language, knowledge of common datasets (e.g., Iris, MNIST), proficiency in using machine learning libraries (e.g., scikit-learn, TensorFlow).

Course Outcome:

1. Implement diverse ML algorithms: linear regression, logistic regression, multinomial logistic regression, SVM, decision trees, MLP.
2. Apply ML techniques to different datasets.
3. Utilize batch gradient descent with early stopping for softmax regression training.
4. Develop neural network models for problem solving.
5. Use TensorFlow for image classification.
6. Implement regression models for fuel efficiency prediction using TensorFlow and Auto MPG dataset.

Course Code	**Course Title**	**Credits**
PSCSP512	**Machine Learning Practical**	**02**
Note: All the Practical's should be implemented using Python and TensorFlow. Link:Python :https://www.python.org/downloads/ TensorFlow :https://www.tensorflow.org/install		
1	Implement Linear Regression (Diabetes Dataset)	
2	Implement Logistic Regression (Iris Dataset)	
3	Implements Multinomial Logistic Regression (Iris Dataset)	
4	Implement SVM classifier (Iris Dataset)	
5	Train and fine-tune a Decision Tree for the Moons Dataset	
6	Train an SVM regressor on the California Housing Dataset	
7	Implement Batch Gradient Descent with early stopping for Softmax Regression	
8	Implement MLP for classification of handwritten digits (MNIST Dataset)	
9	Classification of images of clothing using Tensorflow (Fashion MNIST dataset)	
10	Implement Regression to predict fuel efficiency using Tensorflow (Auto MPG dataset)	

Programme Name: M.Sc. Computer Science (Semester II) **Total Credits:** 04 **College assessment:** 50	**Course Name:** Natural Language Processing **Total Marks:** 100 **University assessment:** 50

Prerequisite: Basic knowledge of programming and Python, understanding of data structures and algorithms, familiarity with probability and statistics.

Course Outcome:

- Understanding the importance and concepts of Natural Language Processing (NLP)
- Applying algorithms available for the processing of linguistic information and computational properties of natural languages.
- Knowledge on various morphological, syntactic, and semantic NLP tasks.
- Introducing various NLP software libraries and data sets publicly available.
- Designing and developing practical NLP based applications

Course Code	Course Title	Total Credits
PSCS513	**Natural Language Processing**	**04**
MODULE - I **Unit 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 Natural Language Toolkit (NLTK), spaCy, genism **Unit 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 Parsing of PCFGs.		**02**
MODULE - II **Unit 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		**02**

Unit 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	

Text Books:

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 NaturalLanguage 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,NitinIndurkhya, and Fred J. Damerau, Pearson; 2nd edition, 2008
4. Foundations of Statistical Natural Language Processing‖, Manning, Christopher and Heinrich, Schutze, MIT Press, 1997

Programme Name: M.Sc. Computer Science (Semester II) **Total Credits:** 02	**Course Name:** Natural Language Processing Practical **Total Marks:** 50 **University assessment:** 50

Prerequisite: Basic knowledge of programming and Python, understanding of data structures and algorithms, familiarity with probability and statistics.

Course Specific Outcome:

- The ability to describe the concepts of morphology, syntax, semantics, discourse & pragmatics of natural language
- Discover various linguistic and statistical features relevant to the basic NLP task, namely, spelling correction, morphological analysis, parts-of-speech tagging, parsing, and semantic analysis
- Assess and Evaluate NLP based systems
- Ability to choose appropriate solutions for solving typical NLP subproblems (tokenizing, tagging, parsing)
- Analyse NLP problems to decompose them inadequate independent components and develop real-life applications

Course Code	**Course Title**	**Credits**
PSCSP514	**Natural Language Processing Practical**	**02**
Note: - The following set of Practical can be performed using any Python Libraries for NLP such as NLTK, spaCy, genism:Link:-https://www.python.org/downloads/		
1	Write a program to implement sentence segmentation and word tokenization	
2	Write a program to Implement stemming and lemmatization	
3	Write a program to Implement a tri-gram model	
4	Write a program to Implement PoS tagging using HMM & Neural Model	
5	Write a program to Implement syntactic parsing of a given text	
6	Write a program to Implement dependency parsing of a given text	
7	Write a program to Implement Named Entity Recognition (NER)	
8	Write a program to Implement Text Summarization for the given sample text	
	CASE STUDIES	
9	Consider a scenario of applying NLP in Customer Service. Design and develop an application that demonstrates NLP operations for working with tasks and data like voice calls, chats, Ticket Data, Email Data. Process the data to understand the voice of the Customer (intent mining, Top words, word cloud, classify topics). Identify issues, replace patterns and gain insight into sales chats.	
10	Consider a scenario of Online Review and demonstrate the concept of sentiment analysis and emotion mining by applying various approaches like lexicon-based approach and rule-based approaches.	
11	Apply NLP in Banking, Financial Services, and Insurance. Design Application to	

	detect frauds and work with SMS data
12	Demonstrate the use of NLP in designing Virtual Assistants. Apply LSTM, build conversational Bots

Programme Name: M.Sc. Computer Science (Semester **II)**	**Course Name:** Simulation and Modelling
Total Credits: 02	**Total Marks:** 50
College assessment: 25	**University assessment:** 25

Prerequisite: Basic understanding of probability and statistics, familiarity with mathematical modeling concepts, knowledge of programming fundamentals.

Course Outcome:

- To introduce students to the fundamental concepts and components of computer simulation and modeling.
- To provide students with a comprehensive understanding of statistical models commonly used in simulation studies.
- To familiarize students with different queueing models and their characteristics for analyzing system performance.
- To develop students' skills in generating random numbers and random variates for simulation experiments.
- To equip students with the knowledge and techniques for input modeling and selecting appropriate input models for simulations.

Course Code	Course Title	Total Credits
PSCS515	**Simulation and Modelling**	**02**
MODULE - I **Unit 1: Introduction to Simulation and Statistical Models** **Introduction to Simulation:** System and System environment, Components of system, Type of systems, Type of models, Steps in simulation study, Advantages and Disadvantages of simulation. **General Principles:** Concepts of discrete event simulation, List processing Statistical Models in Simulation: Useful statistical model, Discrete distribution, Continuous distribution, Poisson process, Empirical distribution. **Queueing Models:** Characteristics of Queueing systems, Queueing notations, Long run measures of performance of Queueing systems, Steady state behavior of infinite population Markovian models, Steady state behavior finite population model, Network of Queues **Unit 2: Random Number Generation, Random Variate Generation, Input Modeling, and Output Analysis** **Random Number 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, Convolution method,		**02**

Acceptance rejection techniques 9. Input Modeling: Data Collection, Identifying the Distribution of data, Parameter estimation, Goodness of fit tests, Selection input model without data, Multivariate and Time series input models. **Verification and Validation of Simulation Model:** Model building, Verification, and Validation, Verification of simulation models, Calibration and Validation of models **Output Analysis for a Single Model:** Types of simulations with respect to output analysis, Stochastic nature of output data, Measure of performance and their estimation, Output analysis of terminating simulators, Output analysis for steady state simulation	

TextBooks:

1. Discrete Event System Simulation, Jerry Banks, John Carson, Barry Nelson, David Nicol, 3rd Edition, Pearson, 2013
2. Simulation Modeling and Analysis, 5th Edition, McGRAW- HILL, Averill Law, W. David Kelton, 2012

Reference Books:

1. System Simulation, Geoffrey Gordon, Pearson, 2007
2. Theory of Modeling and Simulation, Bernard P. Zeigler, Tag Gon Kim , Herbert Praehofer, Academic Press, 2011
3. System Simulation with Digital Computer, NarsinghDeo, PHI

ELECTIVES

Programme Name: M.Sc. Computer Science (Semester II)	**Course Name: Bioinformatics**
Total Credits: 02	**Total Marks:** 50
College assessment: 25	**University assessment:** 25

Pre-requisite: Basic concepts of Biology, Data Structures and Algorithms, Data Analysis and Visualization.

Course outcomes:

- Gain a solid understanding of fundamental concepts and principles in bioinformatics, including sequence analysis, genome analysis, protein structure prediction, and gene expression analysis
- Acquire the ability to analyze and interpret biological data, such as DNA or protein sequences, microarray data, or next-generation sequencing data
- Learn statistical methods and techniques for extracting meaningful insights from large datasets.
- Develop skills in using bioinformatics tools and software packages commonly used in the field, such as BLAST, EMBOSS and Biopython
- Awareness of Ethical and Legal Considerations
- Develop the ability to stay updated with the latest advancements and emerging trends in bioinformatics research and technologies

Course Code	Course Title	Total Credits
PSCS516a	**Bioinformatics**	**02**
MODULE - I **Unit 1: Biological Data Analysis** **Biological Foundations**: Introduction to molecular biology concepts and terminology, DNA, RNA, and protein structure and function, Genetic variation and mutation **Introduction to Bioinformatics**: Overview of bioinformatics and its applications in biology and medicine, Introduction to biological databases and data formats, Introduction to sequence analysis, structure analysis **Sequence Analysis**: Sequence alignment algorithms (pairwise and multiple sequence alignment), Sequence database searching (BLAST, FASTA), Hidden Markov Models (HMMs) for sequence analysis, Phylogenetic analysis and evolutionary tree construction **Structure Analysis**: Protein structure prediction methods (homology modeling, ab initio methods), Protein structure visualization and analysis tools, Drug discovery **Unit 2: Computational Tools and Methods** **Genomics and Transcriptomics**: Analyzing and manipulating genomic sequences, working with genome annotations and gene features, Analyzing gene expression data		**02**

(RNA-Seq, microarray), Identifying differentially expressed genes **Data Visualization and Reporting**: Visualizing bioinformatics data, Creating interactive visualizations of biological data **Machine Learning and Data Mining in Bioinformatics**: Introduction to machine learning algorithms and techniques, Feature selection and dimensionality reduction in biological data, Predictive modeling for biological data (classification, regression) **Ethical, Legal, and Social Implications**: Ethical considerations in bioinformatics research, Privacy and data security in genomic data, social and policy issues in bioinformatics and personalized medicine	

Text Books:

1. Bioinformatics: Sequence and Genome Analysis by David W. Mount Publisher: Cold Spring Harbor Laboratory Press Publication (4th edition), 2021,
2. Python for Bioinformatics by Tiago Antao, Packt Publishing Publication, 2015
3. Python for Biologists: A complete programming course for beginners" by Martin Jones CreateSpace Independent Publishing Platform, 2013,

Reference Books:

1. Bioinformatics for Beginners: Genes, Genomes, Molecular Evolution, Databases, and Analytical Tools by SupratimChoudhuri, Academic Press Publication, 2014
2. Bioinformatics Programming Using Python: Practical Programming for Biological Data by Mitchell L. Model, O'Reilly Media, 2009

Programme Name: M.Sc. Computer Science (Semester II) **Total Credits:** 02	**Course Name:** Bioinformatics Practical **Total Marks:** 50 **University assessment:** 50

Course Code	Course Title	Credits
PSCSP516a	**Bioinformatics Practical**	**02**

1	Sequence Manipulation ● Read and parse sequence data from files ● Perform basic sequence manipulations (e.g., reverse complement, translation)
2	Sequence Alignment ● Perform pairwise sequence alignment using algorithms like Needleman-Wunsch or Smith-Waterman ● Implement multiple sequence alignment using methods such as ClustalW or MUSCLE
3	Database Searching ● Perform sequence searches against databases (e.g., BLAST or FASTA) ● Retrieve and analyze search results
4	Protein Structure Analysis ● Retrieve protein structures from databases like PDB ● Calculate structural properties (e.g., secondary structure, solvent accessibility) ● Perform structure visualization and analysis
5	Genomic Data Analysis ● Retrieve genomic data from databases (e.g., NCBI) ● Analyze gene annotations, promoter regions, or regulatory elements ● Perform genomic variant analysis
6	Data Preprocessing ● Cleaning and preprocessing biological data (e.g., gene expression data, DNA sequences) ● Handling missing values, outliers, and normalization of data ● Feature selection and dimensionality reduction techniques
7	Classification ● Applying machine learning algorithms (e.g., decision trees, random forests, support vector machines) to classify biological samples or sequences ● Evaluating model performance using metrics such as accuracy, precision, recall, and F1-score
8	Regression ● Building regression models to predict quantitative biological properties (e.g., protein structure, gene expression levels) ● Assessing model performance using metrics such as mean squared error or R-squared
9	Clustering ● Applying clustering algorithms (e.g., k-means, hierarchical clustering) to group

	similar biological samples or sequences - Assessing clustering quality using metrics such as silhouette coefficient or Rand index
10	Visualizing clusters and analyzing their biological significance - Data Visualization: - Generate plots, graphs, and figures to visualize bioinformatics results - Use libraries like Matplotlib, Seaborn, or ggplot in Python or R for visualization - Create interactive visualizations using tools like D3.js or Plotly

Programme Name: M.Sc. Computer Science (Semester II)	**Course Name:** Embedded and IoTTechnology
Total Credits: 02	**Total Marks:** 50
University assessment: 25	**College assessment:** 25

Pre requisite: Basic electronics knowledge (components, microcontrollers), understanding of wireless sensor networks, familiarity with IoT concepts and architectures.

Course Outcome:

- Understand embedded systems design and basics of IoT components.
- Learn about electronics and microcontrollers in embedded systems.
- Gain knowledge of IoT building blocks, sensors, and wireless sensor networks.
- Explore advanced IoT technologies, including gateway architecture and cloud computing.
- Understand IoT security, communication, and design connected IoT systems using Packet Tracer.

Course Code	Course Title	Total Credits
PSCS516b	**Embedded and IoT Technology**	**04**
MODULE - I **Unit 1:Embedded System Basics** **Introduction to Embedded Systems-** Design of Embedded Systems, Memory Architecture, Input/Output. Basic electronics: Semiconductors, Transistors, BJT, Flip Flops, Resistors, Capacitors, CMOS, MOSFET, FPGA, Relays. Microcontrollers, UART Communications, SPI-peripherals interface, I2C communication, Wireless Sensor Network (WSN) **Basics of IOT-** Introduction IoT, IoT Building Blocks -Hardware and Software: The basic IoT building blocks, smart thing components and capabilities, basics of Packet Tracer with reference to IoT, basics of IoT gateway, Cloud, and analytics Sensing Principles and Wireless Sensor Network: Sensor fundamentals and classification of sensors, physical principles of some common sensors, basics of WSNs, WSN architecture and types, layer-level functionality of WSN protocol stack. **Unit 2:Advanced IOT Technologies** **IoT Gateway**: IoT architecture domains, IoT gateway architecture, IoT gateway functionalities, IoT gateway selection criteria, IoT gateway and edge computing, edge computing-based solution for specific IoT applications IoT Protocol Stack, IoT Cloud and Fog Computing: Components of IoT Cloud architecture, usage of application domains of IoT Cloud platforms, layered architecture of Fog computing, distinguish Fog computing from other related terms IoT Applications: Main applications of IoT, Implementation details of various IoT application domains **Security, Communication and Data analytics in IOT-**IoT Security: Security constraints		**02**

in IoT systems, security requirements of IoT systems, IoT attacks, security threats at each layer of IoT architecture, design secure IoT system for specific application Social IoT: Nature of social relationships among IoT Devices, functionality of different components of social IoT architecture, social aspects of smart devices in IoT applications Packet Tracer and IoT: Basics of Packet Tracer and Blockly programming language, design simple IoT projects in Packet Tracer.	

Text Books:

1. Introduction to Embedded Systems – Cyber physical systems Approach Edward Ashford Lee &SanjitArunkumarSeshia Second Edition — MIT Press — 2017
2. Enabling the Internet of Things Fundamentals, Design and Applications by Muhammad Azhar Iqbal, SajjadHussain, Huanlai Xing, Muhammad Ali Imran Wiley Pub.1st Edition 2021

Reference Books:

1. Introduction Embedded Systems by K.V. Shibu Second Edition McGraw Hills–2017
2. Build your own IoT Platform Develop a Fully Flexible and Scalable Internet of Things Platform in 24 Hours by AnandTamboli, 2019 ,Apress

Programme Name: M.Sc. Computer Science (Semester II) **Total Credits:** 02	**Course Name:** Embedded and IoT Technology Practical **Total Marks:** 50 **College assessment:** 50

Pre requisite: Knowledge of Embedded Systems.

Course Outcome: -

- The course is designed to enable students, to understand and implement IoT in industry.
- Design and executive projects in IoT with Automatic Identification and Data Capture.

Course Code	Course Title	Credits
PSCSP516b	**Embedded and IoT Technology Practical**	**02**

Note: - The following set of practicals should be implemented in CodeVisionAVR, Proteus8, Cisco Packet Tracer, Keli V5, Python
Link: -Python:https://www.python.org/downloads/
CodeVisionAVR :https://www.codevision.be/
Proteus8:https://www.labcenter.com/downloads/
Cisco Packet Tracer:https://www.netacad.com/courses/packet-tracer
Keli V5: https://www.keil.com/download/

1	Design and implement basics embedded circuits 1. Automatic Alarm system- Alarm should get trigger by senor 2. Timer based buzzer 3. Sensor based Counting device
2	Demonstrate communication between two embedded devices using UART port
3	Built an IoT system to send ticket before entering the bus.
4	Demonstrate an IoT based game which can be played between two player who are physically at a considerable distance.
5	Develop a IoT application which will record the movement and orientation of your phone and give the data back to the PC
6	Develop an IoT application that will raise an alarm whenever with going to rain outside based on the weather prediction data.
7	Deploy an IoT application which will alert you by beeping or vibrating your phone whenever you get someone call your name.
8	Develop an IoT application for monitoring water levels in tanks and automatically start the motor to fill the tank if the level goes below the critical level.
9	Develop an IoT module to which measure the intensity of light and send the same to your PC/ Phone
10	Develop an IoT application for Motion detection.

Programme Name: M.Sc. Computer Science (Semester II)	**Course Name:** Web Data Analytics
Total Credits: 02	**Total Marks:** 50
College assessment: 25	**University assessment:** 25

Prerequisite: Data mining Techniques. Knowledge of Python for implementation.

Course Outcome:

- Understand the concepts and techniques of web mining, including sequential pattern mining and rule generation.
- Gain knowledge of information retrieval models, text preprocessing, and web search techniques.
- Learn about opinion mining and sentiment classification in web information retrieval.
- Explore social network analysis, link analysis, and the implementation of webpage crawlers.
- Understand web usage mining, including the discovery and analysis of web usage patterns, and the use of recommender systems and query log mining.

Course Code	Course Title	Total Credits
PSCS516c	**Web Data Analytics**	**02**
MODULE - I **Unit 1: Introduction to Web Mining** **Web Mining**-Data Mining, Basic Concepts, Difference, Mining Sequential Patterns on Prefix Span, Generating Rules from Sequential Patterns. Basic Concepts of Information Retrieval, Information Retrieval Models, Relevance feedback, Evaluation measures Text and Web Page Preprocessing, Inverted Index and Its Compression, latent semantic indexing, Web Search, Web Spamming **Opinion Mining and Web Usage Mining:**Web Information Retrieval, Sentiment Classification, Feature based Opinion Mining and summarization, Comparative Sentence and Relation Mining, Opinion Search and Opinion Spam. Web Usage Mining. **Unit 2:Social Network & Link Analysis** **Social Network-**Link Analysis, Scrapy using python (without pipelining), Social Network Analysis, Co-Citation and Bibliographic Coupling, PageRank, HITS, Community Discovery **Webpage crawlers and usage mining:** Basic Crawler Algorithm, Implementation Issues, Universal Crawlers, Focused Crawlers, Topical Crawlers, Crawler Ethics and Conflicts, Data modelling and webpage usage mining., Discovery and analysis of web usage patterns, Recommender systems and collaborative filtering, query log mining		**02**

Text Books:

1. Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data by Bing Liu (Springer Publications) 2017 publication
2. Data Mining: Concepts and Techniques, Second Edition Jiawei Han, MichelineKamber (Elsevier Publications),2017

Reference Books:

1. Web Mining: Applications and Techniques by Anthony Scime,2010
2. Mining the Web: Discovering Knowledge from Hypertext Data by SoumenChakrabarti 2010

Programme Name: M.Sc. Computer Science (Semester II) **Total Credits:** 02	**Course Name:** Web Data Analytics Practical **Total Marks:** 50 **University assessment:** 50

Prerequisite: Basic understanding of data mining concepts, familiarity with programming (Python), and knowledge of web technologies (HTML, HTTP).

Course Outcome:

- Develop deep understanding of mining techniques exclusively for the Internet
- Understand and develop analytics for social media data.
- Design and implementation of various web analytical tool to understand complex unstructured data on the Internet for aiding individuals and Businesses to grow their business

Course Code	**Course Title**	**Credits**
PSCSP516c	**Web Data Analytics Practical**	**02**

Note: - The following set of practical's should be implemented in Scrape, python:
Link:-Python : https://www.python.org/downloads/

1	Scrape an online E-Commerce Site for Data. 1. Extract product data from Amazon - be it any product and put these details in the MySQL database. One can use pipeline. Like 1 pipeline to process the scraped data and other to put data in the database and since Amazon has some restrictions on scraping of data, ask them to work on small set of requests otherwise proxies and all would have to be used. 2. Scrape the details like color, dimensions, material etc. Or customer ratings by features
2	Scrape an online Social Media Site for Data. Use python to scrape information from twitter.
3	Page Rank for link analysis using python Create a small set of pages namely page1, page2, page3 and page4 apply random walk on the same
4	Perform Spam Classifier
5	Demonstrate Text Mining and Webpage Pre-processing using meta information from the web pages (Local/Online).
6	Apriori Algorithm implementation in case study.
7	Develop a basic crawler for the web search for user defined keywords.
8	Develop a focused crawler for local search.
9	Develop a programme for deep search implementation to detect plagiarism in documents online.
10	Sentiment analysis for reviews by customers and visualize the same.

Programme Name: M.Sc. Computer Science (Semester II)	**Course Name:** On Job Training
Total Credits: 04	**Total Marks:** 100
College assessment: 50	**University assessment:** 50

A. Introduction

- On Job training (OJT) is an integral component of the M.Sc. Computer Science program that provides students with a unique opportunity to bridge the gap between theoretical knowledge gained in the classroom and practical application in a real-world environment. This training aims to equip students with both technical and non-technical skills that are essential for success in the industry.
- By participating in OJT, students are able to apply the concepts and theories learned during their coursework to real-world scenarios. They gain hands-on experience, problem-solving skills, and a deeper understanding of how the industry operates. This practical exposure enhances their competence and confidence, preparing them to tackle the challenges they may encounter in their professional careers.
- From an organizational perspective, hosting OJT programs allows companies to gain insights into the curriculum and content of the M.Sc. Computer Science program. They can provide valuable feedback on the relevance of the coursework and industry requirements, enabling academic institutions to continually improve the program's alignment with industry needs. This collaboration between academia and industry fosters a mutually beneficial relationship, ensuring that graduates are well-prepared for the job market.
- Moreover, OJT benefits the faculty members involved in the program. They have the opportunity to gain firsthand exposure to the industry and observe the type of work being performed. This experience enables them to enhance their teaching methodologies and delivery techniques, ensuring that they remain up-to-date with the latest industry practices. The insights gained from OJT also enable faculty members to provide relevant guidance and mentorship to students, preparing them for successful careers in the field of computer science.

B. Enhancing Practical Skills through OJT

- The OntheJob Training (OJT) program spans 4-6 weeks, requiring a minimum of 120 hours of physical presence at the organization.
- Students are expected to find their own OJT placements, although the institution provides support and guidance in securing positions with reputable organizations.
- OJT must be conducted outside the home institution to expose students to real-world work environments.

- OJT covers any subject within the syllabus, allowing students to align their experience with their academic interests.
- In recognition of changing dynamics, some OJT sessions can be conducted online to accommodate virtual work environments.
- OJT will offer students the opportunity to apply classroom learning in a real-world setting, fostering the development of technical and non-technical skills.
- Mutual Benefits: Organizations gain insights into the program's curriculum and industry requirements, enabling them to provide constructive feedback and enhance course relevance.
- OJT bridges the gap between theoretical knowledge and practical application, preparing students for successful careers in computer science

C. **Interning organization:**Students have the flexibility to pursue their OJT in various types of organizations, including but not limited to:

- Software Development Firms: Gain practical experience in software development and programming.
- Hardware/Manufacturing Firms: Learn about hardware design, manufacturing processes, and quality assurance.
- Small-Scale Industries/Service Providers: Explore opportunities in diverse sectors such as banking, clinics, NGOs, and professional institutions like CA firms or law firms.
- Civic Departments: Engage with local civic departments such as ward offices, post offices, police stations, or panchayats to understand their functioning and contribute to their activities.
- Research Centre’s/University Departments/Colleges: Contribute as research assistants or in similar roles for research projects or initiatives, fostering collaboration between academia and industry.

Note: The listed options provide a range of possible OJT placements, offering students valuable exposure to different sectors and professional settings.

D. **OJT mentors:**

To enhance the learning experience and ensure the quality of the MSc program, each student participating in the OJT will be assigned two mentors: a faculty mentor from the institution and an industry mentor from the organization where the student is interning.

- **Industry Mentor Role:** The industry mentor plays a crucial role in guiding the student during the internship. They ensure that the internee fulfills the requirements of the organization and successfully meets the demands of the assigned project. Through their expertise and experience, industry mentors provide valuable insights into real-world practices and industry expectations.

- **Faculty Mentor Role:** The faculty mentor serves as the overall coordinator of the OJT program. They oversee the entire internship process and evaluate the quality of the OJT in a consistent manner across all students. The faculty mentor ensures that the OJT aligns with the program's objectives and provides valuable learning opportunities. They also facilitate communication between the institution, industry mentor, and student to ensure a fruitful OJT experience.

By having both an industry mentor and a faculty mentor, students benefit from a comprehensive guidance system that combines industry expertise and academic support.

E. Submission of documentation for OJT

The student will make two documents as part of the OJT

- Online diary: This ensures that the student updates daily activity, which could be accessed by both the mentors. Daily entry can be of 3- 4 sentences giving a very brief account of the learning/activities/interaction taken place. The faculty mentor will be monitoring the entries in the diary regularly as shown in **Appendix-I**
- OJT report: A student is expected to make a report based on the OJT he or she has done in an organization. It should contain the following:
    - **Certificate**: A certificate in the prescribed Performa (given **in Appendix II** and **Appendix III**) from the organization where the OJT was done.
    - **Title**: A suitable title giving the idea about what work the student has performed during the OJT.
    - **Description of the organization**: A small description of the organization where the student has interned
    - **Description of the activities** done by the section where the intern has worked: A description of the section or cell of the organization where the intern 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 allottedand done by the intern:** A detailed description of the work allotted, and actual work performed by the intern during the OJT period. It shall be the condensed and structured version of the daily report mentioned in the online diary.
    - **Self-assessment**: A self-assessment by the intern on what he or she has learned during the OJT period. It shall contain both technical as well as interpersonal skills learned in the process.

F. Interaction between mentors:

To ensure the smooth conduct of the OJT a meet-up involving the intern, industry mentor, and the faculty mentor will be scheduled as a mid-term review. The meeting can preferably be online to save time and resources. The meeting ensures the synergy between all stakeholders of the OJT. A typical meeting can be of around 15 minutes where at the initial stage the intern brief about the work and interaction goes for about 10 minutes. This can be followed by the interaction of the mentors in the absence of the intern. This ensures that issues between the intern and the organization, if any, are resolved amicably.

G. OJT workload for the faculty:

Every student is provided with a faculty member as a mentor. So, a faculty mentor will have a few students under him/her. A faculty mentor is the overall in charge of the OJT of the student. He/she constantly monitors the progress of the OJT by regularly overseeing the diary, interacting with the industry mentor, and guiding on the report writing etc. Considering the time and effort involved, a faculty mentor who is in-charge of 20 students shall be provided by a workload of 3 hours.

EVALUATION SCHEME

A. Evaluation for Mandatory Theory Courses (4 Credit Courses)

I. Internal Evaluation for Mandatory Theory Courses – 50 Marks

(i) Mid-Term Class Test – 30Marks
(ii) Assignment/ Case study– 20 Marks

OR

(i) SWAYAM (Advanced Course) of minimum 20 hours and certification examcompleted – 50 Marks

OR

(ii) NPTEL (Advanced Course) of minimum 20 hours and certification examcompleted - 50 Marks

OR

(iii)Valid International Certifications (Prometric, Pearson, Certiport, Coursera,Udemy and the like) - 50 Marks

One certification marks shall be awarded one course only. For four courses, the students will have to complete four certifications.

II. External Examination for Mandatory Theory Courses – 50 Marks

- Duration: **2.0 Hours**
- Theory question paper pattern:

	All questions are compulsory.		
Question	**Based on**	**Options**	**Marks**
Q.1	Unit I	*Any 2 out of 4*	10
Q.2	Unit II	*Any 2 out of 4*	10
Q.3	Unit III	*Any 2 out of 4*	10
Q.4	Unit IV	*Any 2 out of 4*	10
Q.5	Unit I, II,III & IV	*Any 2 out of 4*	10

B. Evaluation for Elective Theory Courses (2 Credit Courses)

I. Internal Evaluation for Elective Theory Courses – 25 Marks

(i) Mid-Term Class Test – 15 Marks

(ii) Assignment/ Case study– 10 Marks

II. External Examination for Elective Theory Courses – 25 Marks

- Duration: **1 Hour**
- Theory question paper pattern:

	All questions are compulsory.		
Question	**Based on**	**Options**	**Marks**
Q.1	Unit I	*Any 2 out of 4*	10
Q.2	Unit II	*Any 2 out of 4*	10
Q.3	Unit I & II	*Any 1 out of 2*	5

C. Evaluation for Mandatory & Elective Practical Courses (2 Credit Courses)

- Each Practical Course carries 50 Marks
  - **40 marks + 05 marks (journal) + 05 marks (viva)**
- Duration: **2 Hours**for each practical course.
- Minimum **80% practical** from each core subjects are required to be completed.
- Certified Journal is compulsory for appearing at the time of Practical Exam

D. Evaluation of On Job Training Course (4 Credit Course)

Internal Evaluation	
Online diary	25
Mid-term interaction	25
Total	**50**
External Evaluation	
OJT Documentation	25
Quality & Relevance	10
OJT Viva	15
Total	**50**

APPENDIX-I

Maintain the weekly online diary for each week in the following format.

	Day	**Date**	**Name of the Topic/Module Completed**	**Remarks**
1st WEEK	MONDAY			
	TUESDAY			
	WEDNESDAY			
	THRUSDAY			
	FRIDAY			
	SATURDAY			

Signature of the Faculty mentor: ___________

Seal of the University/College

APPENDIX-II

(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 his/her M.Sc. course in Computer Science of University ofMumbai. 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 details:

Email:

(Seal of the organization)

APPENDIX-III

(Proforma for the Evaluation of the intern by the industry mentor /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 to 5 scale by putting √ in the respective cells]

No	Particular	Excellent	Very Good	Good	Moderate	Satisfactory
1	Attendance & Punctuality					
2	Ability to work in a team					
3	Written and oral communication skills					
4	Problem solving skills					
5	Ability to grasp new concepts					
6	Technical skill in terms of technology, programming etc					
7	Ability to complete the task					
8	Quality of overall work done					

Comments:

____

Signature: ______________________________

Name :

Designation:

Contact details:

Email :

(**Seal of the organization)**

Letter Grades and Grade Points

Semester GPA/ Program CGPA Semester/Program	% of Marks	Alpha-Sign / Letter Grade Result
9.00-10.00	90.0-100	O (Outstanding)
8.00-<9.00	80.0-<90.0	A+ (Excellent)
7.00-<8.00	70.0-<80.0	A (Very Good)
6.00-<7.00	60.0-<70	B+ (Good)
5.50-<6.00	55.0-<60.0	B (Above Average)
5.00-<5.50	50.0-<55.0	C (Average)
4.00-<5.00	40.0-<50.0	P (Pass)
Below 4.00	Below 40	F (Fail)
Ab (Absent)	-	Absent

Appendix-B

Justification for M.Sc. (Computer Science)

1.	Necessity for starting the program:	The technology industry is continuously evolving, and there is a growing demand for skilled computer science professionals with advanced knowledge. Offering an MSc program can help address this demand by producing graduates with specialized skills and expertise in various computer science domains such as artificial intelligence, data science, cyber security, software engineering, etc. Specialization can open up unique career opportunities and increase expertise in a chosen field.
2.	Whether the UGC has recommended the program:	**Yes**
3.	Whether all the programs have commenced from the academic year 2023-24	**Yes**
4.	The programs started by the University are self-financed, whether adequate number of eligible permanent faculties are available?	**Yes** Guest Faculty / Adjunct Professors/ IT Professionals will be invited
5.	To give details regarding the duration of the program and is it possible to compress the program?	2 years. Not possible Its Four semester Course, introduction, and learning research skill (Research Methodology) are taught in the first semester; Problem solving skills and On-Job- Training are the part of the second semester. The third semester and fourth semester comprise of Advanced concepts and completing are search project. Exit is available after first year. Corresponding credits should be earned by the learners.
6.	The intake capacity of each program and no. of admissions given in the current academic year:	120 seats 2023-2024 admission starts from July
7.	Opportunities of Employability / Employment available after undertaking these courses:	MSc in Computer Science can open up a wide range of opportunities for employability and employment. The technology industry is diverse and continually evolving, offering numerous career paths for computer science graduates. Here are some common opportunities available after completing such courses: Software Developer/Engineer, Data Scientist/Analyst,

		Machine Learning Engineer, Cyber security Analyst/Consultant, Research Scientist (in academia/industry), IT Consultant. These are just a few examples, and the field of computer science offers many more opportunities. Additionally, as technology continues to advance, new roles and specializations are constantly emerging, creating a dynamic and promising job market for computer science graduates.

Garje

DrJyotshnaDongardive
Head, Department of Computer Science University of Mumbai

DrShivramGarje
Dean, Science and Technology
University of Mumbai