RAMAIAH
Institute of Technology

CURRICULUM

for the Academic year 2024 – 2025

COMPUTER SCIENCE AND ENGINEERING

VII & VIII SEMESTER B.E.

RAMAIAH INSTITUTE OF TECHNOLOGY
(Autonomous Institute, Affiliated to VTU)
Bangalore – 560054.

About the Institute:

Dr. M. S. Ramaiah a philanthropist, founded 'Gokula Education Foundation' in 1962 with an objective of serving the society. M S Ramaiah Institute of Technology (MSRIT) was established under the aegis of this foundation in the same year, creating a landmark in technical education in India. MSRIT offers 18 UG programs and 13 PG programs. All these programs are approved by AICTE. All eligible UG and PG programs are accredited by National Board of Accreditation (NBA). The institute is accredited with **'A^{+}' grade by NAAC in March 2021** for 5 years. University Grants Commission (UGC) & Visvesvaraya Technological University (VTU) have conferred Autonomous Status to MSRIT for both UG and PG Programs since 2007. The institute has also been conferred autonomous status for Ph.D. program since 2021. The institute is a participant to the Technical Education Quality Improvement Program (TEQIP), an initiative of the Government of India. The institute has 380 competent faculty out of which 70% are doctorates. Some of the distinguished features of MSRIT are: State of the art laboratories, individual computing facility for all faculty members, all research departments active with sponsored funded projects and more than 300 scholars pursuing Ph.D. To promote research culture, the institute has established Centre of Excellence for Imaging Technologies, Centre for Advanced Materials Technology, Centre for Antennas and Radio Frequency Systems (CARFS), Center for Cyber Physical Systems, Schneider Centre of Excellence & Centre for Bio and Energy Materials Innovation. **Ramaiah Institute of Technology has obtained All India Rank 182 in "Scimago Institutions Rankings" for the year 2024.**

The Entrepreneurship Development Cell (EDC) and Section 8 company "Ramaiah Evolute" have been set up on campus to incubate startups. MSRIT has a strong Placement and Training department with a committed team, a good Mentoring/Proctorial system, a fully equipped Sports department, large air-conditioned library with good collection of book volumes and subscription to International and National Journals. The Digital Library subscribes to online e-journals from Elsevier Science Direct, IEEE, Taylor & Francis, Springer Link, etc. The Institute is a member of DELNET, CMTI and VTU E-Library Consortium. The Institute has a modern auditorium, recording studio, and several hi-tech conference halls with video conferencing facilities. The institute has excellent hostel facilities for boys and girls. MSRIT Alumni have distinguished themselves by occupying high positions in India and abroad and are in touch with the institute through an active Alumni Association.

As per the National Institutional Ranking Framework (NIRF), MoE, Government of India, Ramaiah Institute of Technology has achieved 75th rank among 1463 top Engineering Institutions & 21st Rank for School of Architecture in India among 115 Architecture Institutions, for the year 2024.

About the Department

Year of Establishment	1984
Names of the Programmes offered	UG: B.E. in Computer Science and Engineering PG: M.Tech. in Computer Science and Engineering PG: M.Tech. in Computer Networks and Engineering Ph.D M.Sc.(Engg.) by Research

The Department of Computer Science and Engineering (CSE) has eminent emeritus professors, 15 faculties with the doctorate degree and 15 pursuing the doctoral studies. The faculty has been publishing research papers in refereed journals and in conference proceedings. The department also conducts vocational courses and proficiency courses on fundamental and new programming languages and computer science concepts. These courses are conducted beyond college hours'/summer semester by the faculty of the department. Some of the faculty are involved in institutional level activities and actively involved in interdisciplinary research activities. The department has state of the art laboratories like SAP, IBM Centre of Excellence and CUDA learning center. Technical seminars, workshops and hackathons are conducted regularly for UG & PG students. The department encourages the students to conduct and participate in extra-curricular/sports activities. The alumni network is very active and regular meeting are conducted by the department. The department is accredited by Nation Board of Accreditation (NBA). The department has MoUs with leading IT Industries like NVIDIA, SAP, IBM and HP. The department conducts subjects with more of hands- on sessions and encourages students to take up MOOC based online courses in NPTEL, IITBombayX, Coursera, Udacity and edX.

VISION OF THE INSTITUTE

To be an Institution of International Eminence, renowned for imparting quality technical education, cutting edge research and innovation to meet global socio-economic needs

MISSION OF THE INSTITUTE

MSRIT shall meet the global socio-economic needs through

1. Imparting quality technical education by nurturing a conducive learning environment through continuous improvement and customization
2. Establishing research clusters in emerging areas in collaboration with globally reputed organizations
3. Establishing innovative skills development, techno-entrepreneurial activities, and consultancy for socio-economic needs

QUALITY POLICY

We at M. S. Ramaiah Institute of Technology strive to deliver comprehensive, continually enhanced, global quality technical and management education through an established Quality Management System complemented by the synergistic interaction of the stake holders concerned

VISION OF THE DEPARTMENT

To build a strong learning and research environment in the field of Computer Science and Engineering that promotes innovation towards betterment of the society

MISSION OF THE DEPARTMENT

1. To produce Computer Science graduates who, trained in design and implementation of computational systems through competitive curriculum and research in collaboration with industry and research organizations.
2. To educate students in technology competencies by providing professionally committed faculty and staff.
3. To inculcate strong ethical values, leadership abilities and research capabilities in the minds of students so as to work towards the progress of the society.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs):

A B.E (Computer Science & Engineering) graduate of Ramaiah Institute of Technology should, within three to five years of graduation

PEO1 Pursue a successful career in the field of Computer Science & Engineering or a related field utilizing his/her education and contribute to the profession as an excellent employee, or as an entrepreneur.

PEO2 Be aware of the developments in the field of Computer Science & Engineering; Continuously enhance their knowledge informally or by pursuing graduate studies.

PEO3 Be able to work effectively in multidisciplinary and multicultural environments and be responsible members and leaders of their communities.

PROGRAM OUTCOMES (POs):

The Outcomes of the Bachelor of engineering in Computer Science & Engineering Programme are as follows:

Engineering Graduates must be able to:

PO1: **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3: **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4: **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6: **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: **Individual and teamwork:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10: **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: **Life-long learning:** Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOs):

PSO1: Understand the principles, architecture and organization of computers, embedded systems, and computer networks.

PSO2: Apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems that include both hardware and software

PSO3: Apply software design and development practices to develop software applications in emerging areas such as IoT, Data Analytics, Social Networks, Cloud and High-Performance Computing.

Semester-wise Credit Breakdown for B.E Degree Curriculum: Batch 2021-25

Semester / Course Category	First	Second	Third	Fourth	Fifth	Sixth	Seventh	Eighth	Total Credits
Basic Sciences (BSC)	**08**	**08**	**03**	**03**	**--**	**--**	**--**	**--**	**22**
Engineering Sciences (ESC)	**09**	**11**	**--**	**--**	**--**	**--**	**--**	**--**	**20**
Humanities, Social Sciences and Management (HSMC)	**02**	**--**	**01**	**01**	**03**	**03**	**--**	**--**	**10**
Ability Enhancement Course (AEC)	**01**	**01**	**01**	**01**	**01**	**--**	**03**	**--**	**08**
Universal Human Values (UHV)	**--**	**--**	**02**	**--**	**--**	**--**	**--**	**--**	**02**
Professional Core Courses (PCC)	**--**	**--**	**11**	**12**	**11**	**05**	**04**	**--**	**43**
Integrated Professional Core Course (IPCC)	**--**	**--**	**03**	**03**	**03**	**--**	**04**	**--**	**13**
Professional Elective Courses (PEC)	**--**	**--**	**--**	**--**	**03**	**06**	**03**	**--**	**12**
Institutional Open Elective Courses (IOE)	**--**	**--**	**--**	**--**	**--**	**03**	**03**	**--**	**06**
Internship (INT)	**--**	**--**	**--**	**02**	**--**	**02**	**--**	**05**	**09**
Mini Project / Project Work (PW)	**--**	**--**	**--**	**--**	**--**	**03**	**--**	**12**	**15**
Non Credit Mandatory Courses (NCMC)	**--**	**--**	✔	**--**	✔	**--**	**--**	**--**	**--**
Total Credits	**20**	**20**	**21**	**22**	**21**	**22**	**17**	**17**	**160**

SCHEME OF TEACHING VII SEMESTER

Sl. No.	Subject Code	Subject	Teaching Department	Category	Credits				Total contact hours /week
					L	T	P	Total	
1	CS71	Complier Design	CSE	IPCC	3	0	1	4	5
2	CS72	Multicore Architecture and Programming	CSE	PCC	2	1	0	3	4
3	CSE73x	Program Elective Course – 4	CSE	PEC	3	0	0	3	3
4	CSOE0x*	Institutional Open Elective - 2	CSE	IOE	3	0	0	3	3
5	CSL74	Microservices Laboratory	CSE	PCC	0	0	1	1	2
6	CSL75	Skill Enhancement Lab (Data Processing and Visualization with PySpark and Tableau)	CSE	PCC	0	1	2	3	6
Total					**11**	**2**	**4**	**17**	**23**

VII Semester List of Courses for Program Elective Course-4

SI. No	Course Code	Course Name	Credits				Contact Hours
			L	T	P	Total	
1.	CSE731	Quantum Computing	2	0	1	3	4
2.	CSE732	Distributed System	3	0	0	3	3
3.	CSE733	Optimization Techniques	3	0	0	3	3
4.	CSE734	Storage Area Networks	3	0	0	3	3
5.	CSE735	Object Oriented Modelling and Design	2	0	1	3	4

NOTE: *All electives are 3 credits; The Course Teaching Faculty shall define the split up L:T:P in the Lesson plan

Nomenclature: IPCC: Integrated Professional Core Course**, PCC:** Professional Core Course**, IOE:** Institutional Open Elective, **AEC –** Ability Enhancement Course, **PW –** Project Work

L –Lecture, T – Tutorial, P- Practical/ Drawing

Note: CSE73x , where x=1,2,3,4,5,
CSOE0x*, where x=6,7,8,9,… continued from previous

Professional Elective Courses: A professional elective (PEC) course is intended to enhance the depth and breadth of educational experience in Engineering and Technology curriculum. Multidisciplinary courses that are added to supplement the latest trend and advanced technology in the selected stream of engineering. Each group provides an option to select one course out of five courses. The minimum student's strength for offering professional electives is 10. However, this conditional shall not be applicable to cases where the admission to the program is less than 10.

Open Elective Courses:

Students belonging to a particular stream of Engineering and Technology are not entitled for the open electives offered by their parent department. However, they can take an elective offered by other departments, provided they satisfy the prerequisite condition, if any. Registration to open electives shall be documented under the guidance of the Proctor.

Selection of an open elective shall not be allowed if,

The candidate has studied the same course during the previous semesters of the program.

The syllabus content of open electives is similar to that of the Departmental core courses or professional electives.

A similar course, under any category, is prescribed in the higher semesters of the program.

The minimum students' strength for offering open electives is 10. However, this conditional shall not be applicable to cases where the admission to the program is less than 10.

AICTE Activity Points to be earned by students admitted to BE program:

Every regular student, who is admitted to the 4-year degree program, is required to earn 100 activity points in addition to the total credits earned for the program. Students entering 4 years Degree Program through lateral entry are required to earn 75 activity points in addition to the total credits earned for the program. The activity points earned by the student shall be reflected on the students VIII Semester grade card. The activities to earn the points can be spread over the duration of the course. However, minimum prescribed duration should be fulfilled. Activity Points (non-credit) have no effect on SGPA/CGPA and shall not be considered for vertical progression. In case student fail to earn the prescribed activity points; VIII semester Grade Card shall be issued only after earning the required activity Points. Students shall be eligible for the award of degree only after the release of the VIII Semester grade card.

SCHEME OF TEACHING VIII SEMESTER

Sl. No.	Subject Code	Subject	Teaching Department	Category	Credits				Total contact hours /week
					L	T	P	Total	
1	CSP81	Project Work	CSE	PW	0	0	12	12	-
2	INT82	Research / Industrial Internship	CSE	INT	0	0	5	5	-
Total					**0**	**0**	**15**	**17**	
3	PE83	Physical Education		NCMC	All students have to register compulsorily for any one of the courses with the concerned coordinator (Yoga Teacher/ Physical Education Director/ NSS Coordinator) in the beginning of the III semester. Attending the registered course from III to VIII semesters. Qualifying is mandatory for the award of the degree				
	YO83	Yoga							
	NS83	NSS							

Nomenclature: PW: Project Work, **INT –** Internship, **NCMC**: Non-credit Mandatory Course

L –Lecture, T – Tutorial, P- Practical/ Drawing

COMPILER DESIGN	
Course Code: CS71	**Credits: 3:0:1**
Prerequisite: Finite Automata and Formal Languages	**Contact Hours: 42L+14P**
Course Coordinator: Dr. Parkavi. A	

Course Contents

Unit I

Introduction: The Structure of Compilers, **Lexical analysis:** The Role of Lexical Analyzer, Input Buffering, Specifications of Tokens, And Recognition of Tokens.

- Pedagogy / Course delivery tools: Chalk and talk, Power point presentation, Videos
- Lab Component / Practical Topics: Design of lexical analyzers
- Link: https://youtu.be/trocRZqxZFM?si=bRvm1FWe4cdJ-FdX

Unit II

Parsing: Top-down Parsing, Bottom-up Parsing, LR Parsing: SLR parser, Canonical parser, LALR parser.

- Pedagogy / Course delivery tools: Chalk and talk, Power point presentation, Videos
- Lab Component / Practical Topics: Design of Parsers
- Link: https://youtu.be/1qOMlqE6LhU?si=C0ZgcGqJ2fUr8QOR

Unit III

Syntax-Directed Definitions: Evaluation order for SDDs, Applications of Syntax-directed translation, Syntax-directed translation schemes.

- Pedagogy / Course delivery tools: Chalk and talk, Power point presentation, Videos
- Lab Component / Practical Topics: Design of SDTs
- Link: https://youtu.be/MO5MGYm4v0o?si=Tpkybs9GGYvxFVwt

Unit IV

Intermediate Code Generation: Variants of syntax trees, Three-address code, Types and declarations, Translation of expressions, Type checking, Control flow, Switch statements, Intermediate code for procedures.

- Pedagogy / Course delivery tools: Chalk and talk, Power point presentation, Videos
- Lab Component / Practical Topics: Design of SDTs to check types and perform evaluations, ICG generations
- Link: https://youtu.be/EpAzj7zXrbk?si=DNaJSjfOsVZ5RaH_

Unit V

Code Generation: Issues in the design of Code Generator, The Target language, Addresses in the target code, Basic blocks and Flow graphs, Optimization of basic blocks, A Simple Code Generator.

- Pedagogy / Course delivery tools: Chalk and talk, Power point presentation, Videos
- Lab Component / Practical Topics: Basic block generations, design of Optimizers, design of code generators
- Link: https://youtu.be/G1qRCb1RoVc?si=4zfzWrEq8hPUM-5L

Text Books:

1. Alfred V Aho, Monica S. Lam, Ravi Sethi, Jeffrey D Ullman: Compilers- Principles, Techniques and Tools, 2nd Edition, Pearson education, 2012.

Reference Books:

1. Kenneth C Louden: Compiler Construction - Principles & Practice, First Edition, Brooks/Cole, CENGAGE learning, 1997.
2. Andrew W Appel: Modern Compiler Implementation in C, First Edition, Cambridge University Press, 2010.

Course Outcomes (COs):

At the end of the course students should be able to:

1. Construct lexical analyzer to recognize inputs using patterns. (PO1,2,4, PSO-1)
2. Devise different types of syntax analyzers using grammars. (PO1,2,4, PSO-1,2)
3. Illustrate syntax-directed translation schemes for grammars. (PO1,3,4,9, PSO-1)
4. Formulate intermediate code generators for programming statements. (PO1,4, PSO-1)
5. Develop assembly language code for the given optimized intermediate codes. (PO1,2,3,4,5, PSO-1,2)

Course Assessment and Evaluation:

Continuous Internal Evaluation (CIE) : 50 Marks		
Assessment Tools	**Marks**	**Course Outcomes (COs) addressed**
Internal Test-I (CIE-I)	30	CO1, CO2 & CO3
Internal Test-II CIE-II)	30	CO3,CO4 & CO5
Average of the two CIE will be taken for 30 marks		
Other Components		
Literature study & Technical paper writing/Practical assignment & Report	20	CO1, CO2, CO3,CO4 & CO5
The Final CIE out of 50 Marks = Average of two CIE tests for 30 Marks+ Marks scored in Literature study and technical paper writing /coding		
Semester End Examination (SEE)		
Course End Examination (Answer One full question from each Unit-Internal Choice)	100	CO1, CO2, CO3, CO4 & CO5

MULTICORE ARCHITECTURE AND PROGRAMMING	
Course Code: CS72	**Credits: 2:1:0**
Prerequisite: Computer Organization and Unix System Programming	**Contact Hours: 28L+14T**
Course Coordinator: Dr. Mallegowda M	

Course Contents

Unit I

Introduction to High–Performance Computers, Memory Hierarchy, CPU Design: Reduced Instruction Set Computers, Multiple–Core Processors, Vector Processors, Parallel Semantics, Distributed Memory Programming.

- Pedagogy / Course delivery tools: Chalk and talk, Power point presentation, Videos
- Lab Component / Practical Topics: OpenMP programs on Processors
- Link: https://archive.nptel.ac.in/courses/106/105/106105033/

Unit II

Programming Shared Address Space Platforms: Thread Basics, Why Threads? The POSIX Thread API, Thread Creation and Termination, Synchronization Primitives in Pthreads, Controlling Thread and Synchronization Attributes, Thread Cancellation, Composite Synchronization Constructs, for Designing Asynchronous Programs, OpenMP: a Standard for Directive Based Parallel Programming.

- Pedagogy / Course delivery tools: Chalk and talk, Power point presentation, Videos
- Lab Component /Practical Topics: OpenMP programs on Threads
- Link:https://archive.nptel.ac.in/courses/106/102/106102163/

Unit III

Programming using the Message-Passing Paradigm: Principles of Message- Passing Programming, The Building Blocks: Send and Receive Operations, MPI: The Message Passing Interface, Topologies and Embedding, Overlapping Communication with Computation, Collective Communication and Computation Operations, Groups and Communicators.

- Pedagogy / Course delivery tools: Chalk and talk, Power point presentation, Videos
- Lab Component / Practical Topics: OpenMP programs on Message- Passing
- Link: https://onlinecourses.nptel.ac.in/noc24_cs63/preview

Unit IV

Introduction: GPUs as Parallel Computers, Architecture of a Model GPU, Why More Speed or Parallelism? Parallel Programming Languages and Models, Overarching Goals. History of GPU Computing: Evolution of Graphics Pipelines, GPU Computing. Introduction to CUDA: Data Parallelism, CUDA Program Structure, A Matrix-Matrix Multiplication Example, Device Memories and Data Transfer, Kernel Functions and Threading.

- Pedagogy / Course delivery tools: Chalk and talk, Power point presentation, Videos
- Lab Component / Practical Topics: CUDA Programming
- Link:https://www.coursera.org/learn/introduction-to-parallel-programming-with-cuda https://onlinecourses.nptel.ac.in/noc20_cs41/preview

Unit V

CUDA Threads: CUDA Thread Organization, Using blockIdx and threadIdx, Synchronization and Transparent Scalability, Thread Assignment, Thread Scheduling and Latency Tolerance. CUDA Memories: Importance of Memory Access Efficiency, CUDA Device Memory Types, A Strategy for Reducing Global Memory Traffic, Memory as a limiting Factor to Parallelism. Performance Considerations: More on Thread Execution, Global Memory Bandwidth, Dynamic Partitioning of SM Resources, Data Perfecting, Instruction Mix, Thread Granularity, Measured Performance and Summary.

- Pedagogy / Course delivery tools: Chalk and talk, Power point presentation, Videos
- Lab Component / Practical Topics: Programming on CUDA
- Link:https://www.coursera.org/learn/introduction-to-concurrent-programming?specialization=gpu-programming,https://nptel.ac.in/courses/106105220

Text Books:

1. Ananth Grama, Anshul Gupta,Vipin kumar, George Karypis Introduction to parallel computing, second edition, Pearson education publishers.
2. David B Kirk, Wen-mei W. Hwu, "Programming Massively Parallel Processors – A Hands-on Approach", First Edition, Elsevier and nvidia Publishers, 2015.

Reference Books:

1. Thomas Rauber and Gudula Runger Parallel Programming for Multicore and cluster systems,Springer International Edition, 2009 .
2. Rubin H Landau, Oregon State University, http://science.oregonstate.edu/ rubin/
3. Hennessey and Patterson Computer Architecture: A quantitative Approach, Morgan Kaufman Publishers
4. Michael J.Quin "Parallel Programming in C with MPI and Open MP", McGraw Hill.
5. Peter S. Pacheco, ―An Introduction to Parallel Programming, Morgan- Kauffman/Elsevier, 2011

Course Outcomes (COs):

At the end of the course, students should be able to:

1. Explain the technologies and architectures for parallel computing. (PO-1, 8,9,10,11, PSO-2)
2. Design and develop parallel programs using OpenMp programming interface. (PO-1,2,3,4,5,8,9,10, PSO-2)
3. Discuss the principles and architecture of message-passing programming. (PO-1,2,3,4,5,8,9,10, PSO-2)
4. Describe Graphical Processing Units and architecture. (PO-1,2,3,4,5,8,9,10, PSO-2)
5. Analyze the features GPUs, their functionalities and also Design parallel applications using CUDA-C. (PO-1,2,3,4,5,8,9,10, PSO-2)

Course Assessment and Evaluation:

Continuous Internal Evaluation (CIE) : 50 Marks		
Assessment Tools	**Marks**	**Course Outcomes addressed**
Internal Test-I (CIE-I)	30	CO1, CO2 & CO3
Internal Test-II CIE-II)	30	CO4 & CO5
Average of the two CIE will be taken for 30 marks		
Other Components		
Case Study Implementation	20	CO1, CO2, CO3 ,CO4 & CO5
The Final CIE out of 50 Marks = Average of two CIE tests for 30 Marks + Marks scored in Case Study with Implementation		
Semester End Examination (SEE)		
Course End Examination (Answer One full question from each Unit-Internal Choice)	100	CO1, CO2, CO3 ,CO4 & CO5

QUANTUM COMPUTING	
Course Code: CSE731	**Credits: 2:0:1**
Prerequisite: Linear Algebra, Python Programming	**Contact Hours: 28L+14L**
Course Coordinator: Dr. Rajarajeswari S	

Course Content

Unit I

Fundamental concepts: Introduction and overview, Introduction to quantum mechanics: Linear algebra, the postulates of quantum mechanics, Quantum Computing software: Introduction to Qiskit, Quantum Qudit simulator, QCAD, programming a quantum computer: The IBMQ, coding a quantum computer using a simulator to carry out basic quantum measurement and state analysis

- Pedagogy / Course delivery tools: Chalk and talk, Power point presentation, Videos.
- Link:https://learning.quantum.ibm.com/course/basics-of-quantum-information/single-systems, https://docs.quantum.ibm.com/start
- Lab Component / Practical Topics: Install Qiskit, Hello world, Build circuits, variational Quantum Eigen solver, demo of other simulator.

Unit II

Quantum correlations: Bell inequalities and entanglement (CHSH, Mermin, and Svetlichny inequalities), Schmidt decomposition, superdense coding, teleportation, density operator, Quantum Computing software: Quack, qasm2circ. Implementation using Qiskit

- Pedagogy / Course delivery tools: Chalk and talk, Power point presentation, Videos.
- Link: https://learning.quantum.ibm.com/course/basics-of-quantum-information/entanglement-in-action
- Lab Component / Practical Topics: Schmidt decomposition, superdense coding

Unit III

Quantum computation: Quantum Ciruits, Controlled Operations, Measurement, universal Quantum gates, summary, simulation. Quantum Fourier transform and its applications: Quantum fourier transform, phase estimation, Applications. Implementation using Qiskit

- Pedagogy / Course delivery tools: Chalk and talk, Power point presentation, Videos.
- Link:https://learning.quantum.ibm.com/course/basics-of-quantum-information/quantum-circuits,
- Lab Component / Practical Topics: UQG, Fourier Transformations.

Unit IV

Quantum Search Algorithms: QSA, Solovay-Kitaev theorem, Deutsch-Jozsa algorithm, Bernstein-Vazirani Algorithm,Simon's algorithm , Shor's algorithm, Grover's algorithm, Quantum counting, Quantum walk search algorithm .Implementation using Qiskit.

- Pedagogy / Course delivery tools: Chalk and talk, Power point presentation, Videos.
- Link:https://learning.quantum.ibm.com/course/fundamentals-of-quantum-algorithms/quantum-algorithmic-foundations,

https://learning.quantum.ibm.com/course/fundamentals-of-quantum-algorithms/quantum-query-algorithms,
https://learning.quantum.ibm.com/course/fundamentals-of-quantum-algorithms/phase-estimation-and-factoring
https://learning.quantum.ibm.com/course/fundamentals-of-quantum-algorithms/grovers-algorithm,

- Lab Component / Practical Topics: Quantum Approximate Optimization Algorithm, Implement all Quantum Optimization algorithms.

Unit V

Quantum Information: Quantum noise and Quantum operations, Distance measures, Quantum Error Correction, Quantum cryptography: Private Key Cryptography, Privacy amplification and information reconciliation, QKD, privacy and coherent information, security of Quantum Key distribution. Implementation using Qiskit.

- Pedagogy / Course delivery tools: Chalk and talk, Power point presentation, Videos.
- Link: https://www.quantumblockchains.io/qkd-protocol-simulation-with-qiskit/
- Lab Component / Practical Topics: Quantum cryptography algorithm

Text Books:

1. M. A. Nielsen &I.Chuang, Quantum Computation and Quantum Information, Cambridge University Press (2016).
2. Eleanor G. Rieffel, and Wolfgang H. Polak, Quantum Computing: A Gentle Introduction, MIT Press, 2014.
3. https://qiskit.org/documentation/

Reference Books:

1. Chris Bernhardt, Quantum Computing for Everyone, The MIT Press, Cambridge, 2020
2. David McMahon, Quantum Computing Explained, Wiley-Interscience, IEEE Computer Society, 2008.
3. Phillip Kaye, Raymond Laflamme et. al., An introduction to Quantum Computing, Oxford University Press, 2007.
4. Martin Laforest, The Mathematics of Quantum Mechanics, University of Waterloo, Quantum Cryptography School for young students.

Course Outcomes (COs):

At the end of the course, students should be able to:

1. Analyze the behavior of basic quantum computation and Simulate basic quantum measurement and state analysis using Qiskit.(PO-1, 2, 3,4, 5, 9,10,12, PSO-2, 3)
2. Elaborate on quantum non-locality and simulation of the density operators. (PO-1, 2, 3,4, 5, 9,10,12, PSO-2, 3)
3. Prove basic facts about quantum information channels and Implement information channels in the quantum circuit model. .(PO-1, 2, 3,4, 5, 9,10,12, PSO-2, 3)
4. Compare, in terms of time complexity, the quantum advantage expected from the quantum algorithms with respect to their classical counterparts. (PO-1, 2, 3,4, 5, 9,10,12, PSO-2, 3)

5. Simulate a simple quantum error-correcting code. (PO-1, 2, 3,4, 5,7,8, 9,10 ,12, PSO-1, 2, 3)

Course Assessment and Evaluation:

Continuous Internal Evaluation (CIE) : 50 Marks		
Assessment Tools	**Marks**	**Course Outcomes addressed**
Internal Test-I (CIE-I)	30	CO1, CO2 & CO3
Internal Test-II CIE-II)	30	CO4 & CO5
Average of the two CIE will be taken for 30 marks		
Other Components		
Case Study Implementation + Report in IEEE Paper format.	20	CO1, CO2, CO3 ,CO4 & CO5
The Final CIE out of 50 Marks = Average of two CIE tests for 30 Marks+ Marks scored in Case Study with Implementation		
Semester End Examination (SEE)		
Course End Examination (Answer One full question from each Unit-Internal Choice)	100	CO1, CO2, CO3 ,CO4 & CO5

DISTRIBUTED SYSTEMS	
Course Code: CSE732	**Credits: 3:0:0**
Prerequisite: Operating System, Computer Network	**Contact Hours: 42L**
Course Coordinator: Dr Sangeetha V	

Course Contents

Unit I

Introduction: Definition, Relation to computer system components, Motivation, Primitives for distributed communication, Synchronous versus asynchronous executions, Design issues and challenges. **A model of distributed computations**: A distributed program, A model of distributed executions, Models of communication networks, Global state of a distributed system, Cuts of a distributed computation, Past and future cones of an event. **Logical time**: Introduction, A framework for a system of logical clocks, Scalar time, Vector time, efficient implementations of vector clocks, Jard–Jourdan's adaptive technique. Relation to parallel multiprocessor/multicomputer systems, Message-passing systems versus shared memory systems

- Pedagogy / Course delivery tools: Chalk and talk, Power point presentation, Videos
- Lab Component / Practical Topics: Implement clock synchronization (logical/physical), Implement concurrent day-time client-server application.
- Link: https://onlinecourses.nptel.ac.in/noc21_cs87/

Unit II

Global state and snapshot recording algorithms: Introduction, System model and definitions, Snapshot algorithms for FIFO channels, Variations of the Chandy–Lamport algorithm, Snapshot algorithms for non-FIFO channels, Snapshots in a causal delivery system, Monitoring global state.
Terminology and basic algorithms: Topology abstraction and overlays, Classifications and basic concepts, Synchronizers, Maximal independent set (MIS), Leader election. Complexity measures and metrics, Program structure, Elementary graph algorithms

- Pedagogy / Course delivery tools: Chalk and talk, Power point presentation, Videos
- Lab Component / Practical Topics: Incrementing a counter in shared memory, Implement non token based algorithm for Mutual Exclusion.
- Link: https://onlinecourses.nptel.ac.in/noc21_cs87/

Unit III

Message ordering and group communication: Message ordering paradigms, Asynchronous execution with synchronous communication, Synchronous program order on an asynchronous system, Group communication, Causal order (CO), Total order, Classification of application-level multicast algorithms. **Termination detection**: Introduction, System model of a distributed computation, Termination detection using distributed snapshots, Termination detection by weight throwing, a spanning-tree based termination detection algorithm, Termination detection in a very general distributed computing model, Termination detection in the atomic computation model. A nomenclature for multicast, Propagation trees for multicast

- Pedagogy / Course delivery tools: Chalk and talk, Power point presentation, Videos

- Lab Component / Practical Topics: Implement termination detection, Implement an exactly-once RPC protocol on top of an asynchronous network, Implement Leader election algorithm.
- Link: https://onlinecourses.nptel.ac.in/noc21_cs87/

Unit IV

Distributed mutual exclusion algorithms: Introduction, Preliminaries, Lamport's algorithm, Ricart–Agrawala algorithm, Singhal's dynamic information-structure algorithm, Lodha and Kshemkalyani's fair mutual exclusion algorithm, Quorum-based mutual exclusion algorithms, Maekawa's algorithm, Agarwal–El Abbadi quorum-based algorithm, Token-based algorithms, Raymond's tree-based algorithm. **Deadlock detection in distributed systems**: Introduction, System model, Preliminaries, Mitchell and Merritt's algorithm for the single resource model, Chandy–Misra–Haas algorithm for the AND model, Chandy–Misra–Haas algorithm for the OR model, Kshemkalyani–Singhal algorithm for the P-out-of- Q model. Models of deadlocks, Knapp's classification of distributed deadlock detection algorithms

- Pedagogy / Course delivery tools: Chalk and talk, Power point presentation, Videos
- Lab Component / Practical Topics: Implement Lamport's Logical Clock, Implement edge chasing distributed deadlock detection algorithm.
- Link: https://onlinecourses.nptel.ac.in/noc21_cs87/

Unit V

Global predicate detection: Stable and unstable predicates, Modalities on predicates, Centralized algorithm for relational predicates, Conjunctive predicates, Distributed algorithms for conjunctive predicates. **Consensus and agreement algorithms**: Problem definition, Overview of results, Agreement in a failure-free system (synchronous or asynchronous), Agreement in (message-passing) synchronous systems with failures, Agreement in asynchronous message-passing systems with failures. **Peer-to-peer computing and overlay graphs**: Introduction, Data indexing and overlays, Unstructured overlays, Chord distributed hash table. Graph structures of complex networks, Scale-free networks.

- Pedagogy / Course delivery tools: Chalk and talk, Power point presentation, Videos
- Lab Component / Practical Topics: Implement multi-threaded client/server processes, Demonstrate process/code migration.
- Link: https://onlinecourses.nptel.ac.in/noc21_cs87/

Text Book:

1. Ajay D. Kshemkalyani, and Mukesh Singhal "Distributed Computing: Principles, Algorithms, and Systems", Cambridge University Press, 2008 (Reprint 2013).

Reference Books:

1. John F. Buford, Heather Yu, and Eng K. Lua, "P2P Networking and Applications", Morgan Kaufmann, 2009 Elsevier Inc.
2. Kai Hwang, Geoffrey C. Fox, and Jack J. Dongarra, "Distributed and Cloud Computing: From Parallel processing to the Internet of Things", Morgan Kaufmann, 2012 Elsevier Inc.

Course Outcomes (COs):

At the end of the course, the students will be able to:

1. Identify the design issues and Challenges in building distributed systems. (PO-1, 3, 4, PSO-1, 3)
2. Explore basic distributed graph algorithms, synchronizers, and recording global state of distributed computation. (PO-3, 4, 9, PSO-1)
3. Analyze ways to achieve various message ordering schemes for detecting termination of a distributed computation. (PO-1, 3, 4, PSO-1, 3)
4. Discuss distributed algorithms to implement Mutual Exclusion and Deadlock detection. (PO-1, 3, 4, 9, PSO-1, 2)
5. Identify Consensus and agreement algorithms and P2P overlay problems (PO-3, 9, PSO-1, 2, 3)

Course Assessment and Evaluation:

Continuous Internal Evaluation (CIE) : 50 Marks		
Assessment Tools	**Marks**	**Course Outcomes addressed**
Internal Test-I (CIE-I)	30	CO1, CO2 & CO3
Internal Test-II CIE-II)	30	CO4 & CO5
Average of the two CIE will be taken for 30 marks		
Other Components		
Case Study with Implementation	20	CO1, CO2, CO3 ,CO4 & CO5
The Final CIE out of 50 Marks = Average of two CIE tests for 30 Marks+ Marks scored in Case Study with Implementation		
Semester End Examination (SEE)		
Course End Examination (Answer One full question from each Unit-Internal Choice)	100	CO1, CO2, CO3 ,CO4 & CO5

OPTIMIZATION TECHNIQUES	
Course Code: CSE733	**Credits: 3:0:0**
Prerequisite: Engineering Mathematics	**Contact Hours: 42L**
Course Coordinator: Veena GS	

Course Contents

Unit I

Introduction: Introduction to optimization, engineering applications of optimization, Formulation of structural optimization problems as programming problems. Optimization Techniques: Classical optimization techniques, single variable optimization, multivariable optimization with no constraints, unconstrained minimization techniques and algorithms constrained optimization solutions by penalty function techniques, Lagrange multipliers techniques and feasibility techniques.

- Pedagogy / Course delivery tools: Chalk and talk, Power point presentation, Videos
- Lab Component / Practical Topics: Nil
- Link: https://onlinecourses.nptel.ac.in/noc21_me10/preview

Unit II

Linear Programming: Linear programming, standard form of linear programming, geometry of linear programming problems, solution of a system of linear simultaneous equations, pivotal production of general systems of equations, simplex algorithms, revised simpler methods, duality in linear programming

- Pedagogy / Course delivery tools: Chalk and talk, Power point presentation, Videos
- Lab Component / Practical Topics: Nil
- Link:https://onlinecourses.nptel.ac.in/noc21_me10/preview

Unit III

Non-linear programming: Non-linear programming, one dimensional minimization methods, elimination methods, Fibonacci method, golden section method, interpolation methods, quadratic and cubic methods, Unconstrained optimization methods, direct search methods, random search methods, descent methods

- Pedagogy / Course delivery tools: Chalk and talk, Power point presentation, Videos
- Lab Component / Practical Topics: Nil
- Link: https://onlinecourses.nptel.ac.in/noc21_me10/preview

Unit IV

Constrained optimization techniques such as direct methods, the complex methods, cutting plane method, exterior penalty function methods for structural engineering problems. Formulation and solution of structural optimization problems by different technique

- Pedagogy / Course delivery tools: Chalk and talk, Power point presentation, Videos
- Lab Component / Practical Topics: Nil
- Link:https://onlinecourses.nptel.ac.in/noc21_me10/preview

Unit V

Geometric programming: Geometric programming, conversion of NLP as a sequence of LP/ geometric programming. Dynamic programming: Dynamic programming conversion of NLP as a sequence of LP/ Dynamic programming

- Pedagogy / Course delivery tools: Chalk and talk, Power point presentation, Videos
- Lab Component / Practical Topics: Nil
- Link: https://onlinecourses.nptel.ac.in/noc21_me10/preview

Text Books:

1. Spunt ,"Optimum Structural Design"- Prentice Hall 2017
2. S.S. Rao, "Optimization – Theory and Practice"- Wiley Eastern Ltd.
3. Uri Krisch, "Optimum Structural Design"- McGraw Hill
4. Richard Bronson, "Operation Research"- Schaum's Outline Series
5. Bhavikatti S.S.- "Structural optimization using sequential linear programming"- Vikas publishing house

Course Outcomes (COs):

At the end of the course, the student should be able to:

1. Apply Analytical Techniques to tackle complex engineering optimization problems. (PO-1, 2, 3,4, 5, 9,10,12, PSO-2, 3)
2. Apply their understanding of linear programming principles and techniques to formulate real-world optimization problems. (PO-1, 2, 3,4, 5, 9,10,12, PSO-2, 3)
3. Demonstrate the ability to proficiently handle non-linear optimization challenges. (PO-1, 2, 3,4, 5, 9,10,12, PSO-2, 3)
4. Demonstrate proficiency in formulating and resolving structural optimization challenge. (PO-1, 2, 3,4, 5, 9,10,12, PSO-2, 3)
5. Understands the concept of Dynamic programming to solve complex problems in various domains. (PO-1, 2, 3,4, 5, 9,10,12, PSO-2, 3)

Course Assessment and Evaluation:

Continuous Internal Evaluation (CIE) : 50 Marks		
Assessment Tools	**Marks**	**Course Outcomes addressed**
Internal Test-I (CIE-I)	30	CO1, CO2 & CO3
Internal Test-II CIE-II)	30	CO4 & CO5
Average of the two CIE will be taken for 30 marks		
Other Components		
Case Study Implementation	20	CO1, CO2, CO3 ,CO4 & CO5
The Final CIE out of 50 Marks = Average of two CIE tests for 30 Marks +Marks scored in Case Study with Implementation		
Semester End Examination (SEE)		
Course End Examination (Answer One full question from each Unit-Internal Choice)	100	CO1, CO2, CO3 ,CO4 & CO5

STORAGE AREA NETWORKS	
Course Code: CSE734	**Credits: 3:0:0**
Prerequisite: Computer Networks, Computer Organization, Operating Systems	**Contact Hours: 42L**
Course Coordinator: Dr. Sangeetha V	

Course Contents

Unit I

Introduction to Information Storage: Information Storage, Evolution of Storage Architecture, Data Centre Infrastructure, Virtualization and Cloud Computing. **Data Centre Environment:** Application, Database Management System, Host (Compute), Connectivity, Storage, Disk Drive Components, Disk Drive Performance, Host Access to Data, Direct-Attached Storage, Storage Design Based on Application, Disk Native Command Queuing, Introduction to Flash Drives.

- Pedagogy / Course delivery tools: Chalk and talk, Power point presentation, Videos
- Lab Component / Practical Topics: Study and Verification of standard Network topologies i.e. Star, Bus, Ring etc.
- Link: https://nptel.ac.in/courses/106108058

Unit II

Data Protection: RAID Implementation Methods, RAID Array Components, RAID Techniques, RAID Levels, RAID Impact on Disk Performance, RAID Comparison, Hot Spares. **Intelligent Storage System**: Components of an Intelligent Storage System, Storage Provisioning, Types of Intelligent Storage Systems.

- Pedagogy / Course delivery tools: Chalk and talk, Power point presentation, Videos
- Lab Component / Practical Topics: RAID installations and Configurations
- Link: https://nptel.ac.in/courses/106108058

Unit III

Fiber Channel Storage Area Networks: Fiber Channel: Overview, The SAN and Its Evolution, Components of FC SAN, FC Connectivity, Switched Fabric Ports, Fibre Channel Architecture, Fabric Services, Switched Fabric Login Types, Zoning, FC SAN Topologies, Virtualization in SAN. **IP SAN and FCoE :** iSCSI (Small Computer Systems Interface), FCIP (Fibre Channel over IP), FCoE (Fibre Channel Over Ethernet).

- Pedagogy / Course delivery tools: Chalk and talk, Power point presentation, Videos
- Lab Component / Practical Topics: Implement Switched Fabric Ports and SAN Topologies.
- Link: https://nptel.ac.in/courses/106108058

Unit IV

Network-Attached Storage: Benefits of NAS, Components of NAS, NAS I/O Operation, NAS Implementations, NAS File Sharing Protocols, Factors Affecting NAS Performance, File-Level Virtualization. **Object Based and Unified Storage**: Object Based Storage Devices, Content Addressed Storage, CAS Use Cases, Unified Storage.

- Pedagogy / Course delivery tools: Chalk and talk, Power point presentation, Videos
- Lab Component / Practical Topics: Implement the NAS and demonstrate file sharing
- Link: https://nptel.ac.in/courses/106108058

Unit V

Introduction to Business Continuity: Information Availability, BC Terminology, BC Planning Lifecycle, Failure Analysis, Business Impact Analysis. **Backup and Archive:** Backup Purpose, Backup Considerations. **Securing the Storage Infrastructure**: Information Security Framework, Risk Triad, Storage Security Domains. **Managing the Storage Infrastructure:** Monitoring the Storage Infrastructure, Storage Infrastructure Management Activities, Storage Infrastructure Management Challenges, Information Lifecycle Management, Storage Tiering.

- Pedagogy / Course delivery tools: Chalk and talk, Power point presentation, Videos
- Lab Component / Practical Topics: Implement security framework
- Link: https://nptel.ac.in/courses/106108058

Text Books:

1. Somasundaram Gnanasundaram, Alok Shrivastava: "Information Storage and Management", 2nd Edition, EMC Education Series, Wiley, Publishing Inc., 2012.

Reference Books:

1. Ulf Troppens, Rainer Erkens and Wolfgang Muller: "Storage Networks Explained', 2nd Edition, John Wiley & Sons Inc, 2009.
2. Robert Spalding: "Storage Networks, the Complete Reference", 1stEdition, Tata McGraw Hill, 2017.

Course Outcomes (COs):

At the end of the course, the student should be able to:

1. Identify the need for storage centric network and its benefits of its adoption. (PO-1,2,3, 5 PSO-1,2)
2. Design a storage solution for an application depending on the IOPS and RAID requirements. (PO-1,2,3,5, 12, PSO-1,2)
3. Examine the working of Fiber Channel Storage Area Networks and IP-SAN. (PO-1,2,3,5,11 PSO-1,2)
4. Select the appropriate Network-Attached Storage and object oriented storage infrastructure to balance between cost and performance. (PO-1,2,3,6, PSO-1,2)
5. Outline the business continuity plan and Secure Management of storage. (PO-1,2,3,12, PSO-1,2)

Course Assessment and Evaluation:

Continuous Internal Evaluation (CIE) : 50 Marks		
Assessment Tools	**Marks**	**Course Outcomes addressed**
Internal Test-I (CIE-I)	30	CO1, CO2 & CO3
Internal Test-II CIE-II)	30	CO4 & CO5
Average of the two CIE will be taken for 30 marks		
Other Components		
Case Study Implementation	20	CO1, CO2, CO3 ,CO4 & CO5
The Final CIE out of 50 Marks = Average of two CIE tests for 30 Marks+ Marks scored in Case study		
Semester End Examination (SEE)		
Course End Examination (Answer One full question from each Unit-Internal Choice)	100	CO1, CO2, CO3 ,CO4 & CO5

OBJECT ORIENTED MODELING AND DESIGN	
Course Code: CSE735	**Credits: 2:0:1**
Prerequisite: Object Oriented Programming, Software Engineering	**Contact Hours: 28L+14P**
Course Coordinator: Dr. Rajarajeswari S	

Course Contents

Unit I

Introduction: object oriented development-modeling concepts, object oriented methodology, models, object oriented themes, Object Modeling, links and associations, advanced links and association concepts, generalization and inheritance, grouping constructs, a sample object model. Advanced Object Modeling: aggregation, abstract classes, generalization as extension and restriction, multiple inheritance, metadata, candidate keys, constraints.

- Pedagogy /Course delivery tools: Chalk and talk, Power point presentation, Videos.
- Link: https://www.ibm.com/support/pages/ibm-rational-rose-enterprise-7004-ifix002, https://www.ibm.com/docs/en/rsm/7.5.0?topic=tours-create-diagrams-using-rational-uml-modeling-tools, https://www.ibm.com/docs/en/rational-soft-arch/9.6.1?topic=diagrams-use-case
- Lab Component / Practical Topics: Create Problem stmt for the case study, using rational rose suite, identify use cases and classes for it.

Unit II

Dynamic modeling: Events and states, Operations Nested state diagrams, Concurrency – Advanced dynamic modeling concepts, A sample dynamic model, Relationship of Object and Dynamic models.

Dynamic Diagrams – UML interaction diagrams, System sequence diagram, Collaboration diagram – When to use Communication Diagrams, State machine diagram and Modeling –When to use State Diagrams, Activity diagram, Implementation Diagrams, UML package diagram.

- Pedagogy /Course delivery tools Chalk and talk, Power point presentation, Videos
- Link: https://www.ibm.com/docs/en/rational-soft-arch/9.7.0?topic=samples-uml-models
- Lab Component /Practical Topics Given case study- UML diagrams implementation using IBM rational Rose suite.

Unit III

Analysis: Analysis in object modeling, dynamic modeling and functional modeling, adding operations, Iterating the analysis. System Design: Breaking system into subsystems, identifying concurrency, allocating subsystems to processors and tasks, managing of data stores. Handling of global resources-handling boundary conditions, Common Architectural Frameworks.

- Pedagogy / Course delivery tools: Chalk and talk, Power point presentation, Videos
- Link : https://www.ibm.com/docs/en/rational-soft-arch/9.7.0?topic=designing-modeling https://www.ibm.com/docs/en/dma?topic=models-using-architecture-frameworks-uml
- Lab Component / Practical Topics: Creating UPIA models, DoDAF model, Integrating UML application design with deployment planning in topologies using rational rose suite.

Unit IV

Object Design: Overview of Object design, Combining the three models, Designing algorithms, Design optimization, Implementation of control, Adjustment of inheritance, Design of association, Object representation, Physical packaging, Documenting design decisions, Comparison of methodologies.

- Pedagogy /Course delivery tools: Chalk and talk, Power point presentation, Videos.
- Link: https://www.ibm.com/docs/en/rational-soft-arch/9.7.0?topic=samples-uml-models
- Lab Component /Practical Topics: IBM rational Rose suite framework for optimizing the Object design.

Unit V

DESIGN PATTERNS – 1: What is a pattern and what makes a pattern? Pattern categories, Relationships between patterns, Pattern description. Communication Patterns: Forwarder-Receiver, Client-Dispatcher-Server, Publisher Subscriber.

DESIGN PATTERNS – 2, IDIOMS: Management Patterns: Command processor, View handler. Idioms: Introduction, what can idioms provide? Idioms and style; Where to find idioms, Counted Pointer example.

- Pedagogy /Course delivery tools: Chalk and talk, Power point presentation, Videos.
- Links: https://www.geeksforgeeks.org/software-design-patterns/
- Lab Component /Practical Topics: Case study solving in java, python/dream weaver

Text Books:

1. Ali Bahrami, - Object Oriented Systems Development‖, Irwin McGraw Hill,2nd edition, 2004.
2. Grady Booch, - Object Oriented Analysis and Design with Applications‖, Pearson Education, 3rd Edition, 2009.
3. Michael Blaha, James Rumbaugh: Object-Oriented Modeling and Design with UML, 2nd Edition, Pearson Education, 2005.
4. Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, Michael Stal: Pattern-Oriented Software Architecture, A System of Patterns, Volume 1, John Wiley and Sons, 2007.

Course Outcomes (COs):

At the end of the course, the student will be able to:

1. Describe the concepts involved in Object-Oriented modelling and their benefits. (PO1,2,3,4,5,9,10, PSO2)
2. Demonstrate concept of use-case model, sequence model and state chart model for a given problem. (PO1,2,3,4,5,9,10, PSO2)
3. Analyze the concept of systems design, modeling and a functional design. (PO1,2,3,4,5,9,10, PSO2)
4. Design class and object diagrams that represent static aspects of a software system. (PO1,2,3,4,5,9,10, PSO2)
5. Apply appropriate design patterns and idioms. (PO1,2,3,4,5,9,10, PSO2)

Course Assessment and Evaluation:

Continuous Internal Evaluation (CIE) : 50 Marks		
Assessment Tools	**Marks**	**Course Outcomes addressed**
Internal Test-I (CIE-I)	30	CO1, CO2 & CO3
Internal Test-II CIE-II)	30	CO4 & CO5
Average of the two CIE will be taken for 30 marks		
Other Components		
Case Study Implementation and report documentation.	20	CO1, CO2, CO3 ,CO4 & CO5
The Final CIE out of 50 Marks = Average of two CIE tests for 30 Marks+ Marks scored in Case Study with Implementation		
Semester End Examination (SEE)		
Course End Examination (Answer One full question from each Unit-Internal Choice)	100	CO1, CO2, CO3 ,CO4 & CO5

MICRO SERVICES LABORATORY	
Course Code: CSL74	**Credits: 0:0:1**
Prerequisite: Java Programming	**Contact Hours: 14P**
Course Coordinator: Dr. Geetha J	

Course Contents

- Overview of Microservices
- Microservice Architecture
- Preparing the Environment
- Introduction to Spring Boot for Microservices
- Getting started with Hello World Microservice
- Building microservices with Spring Boot
- Testing microservices
- Communication between 2 or more microservices through REST services (setting up Eureka service and two or three services)
- Distributed log tracing by using zepkln and spring cloud sleuth. (i.e. tracing a request from multiple microservice)
- Mocking and Testing service by using Mocikto and Junit
- Connecting spring microservices with a data base.
- Setting up Swagger ui for a microservice.
- Creating custom annotations (at method level and class level)
- Deploying microservices using Docker/Kubernetes

Reference Books:

1. Chris Richardson: Microservices Patterns with Examples in Java, Manning Publications Co., First Edition, 2019.
2. Moises Macero: Learn Microservices with Spring Boot: A Practical Approach to RESTful Services using RabbitMQ, Eureka, Ribbon, Zuul and Cucumber, A Press, First Edition, 2017.
3. Sourabh Sharma: Mastering Microservices with Java 9, Packt Publishing Ltd, Second Edition, 2017.

Course Outcomes (COs):

At the end of the course, the student should be able to:

1. Illustrate the importance of Microservices as an Architecture Implementation. (PO-1,2,3, PSO-2)
2. Develop an application on Java with MongoDB and convert over monolithic structure to Micro services. (PO- 1, 2,3, PSO-3)
3. Deploy application on docker and Access the Kubernetes. (PO- 1,2,3,5, PSO-3)

Course Assessment and Evaluation:

Continuous Internal Evaluation (CIE) : 50 Marks		
Assessment Tools	**Marks**	**Course Outcomes addressed**
Internal Test-I (CIE-I)	20	CO1, CO2, CO3
Regularity/ Continuous assessment	10	CO1, CO2, CO3
Other Components		
Microservice Implementation	20	CO1, CO2, CO3
The Final CIE out of 50 Marks = CIE TEST+ Continuous assessment +Project Implementation		
Semester End Examination (SEE)		
Course End Examination	50	CO1, CO2, CO3

SKILL ENHANCEMENT LAB	
Course Code: CSL75	**Credits: 0:1:2**
Prerequisite: Java Programming	**Contact Hours: 14T+28P**
Course Coordinator: Dr. Geetha J	

Course Contents

- Introduction to Big Data and Analytics
- Introduction to PySpark
- PySpark Basics
- Advanced PySpark Techniques
- PySpark Machine Learning
- Introduction to Tableau
- Advanced Tableau Techniques
- Integrating PySpark with Tableau
- Creating Visualizations from PySpark Data and Building Dashboards using PySpark Data)
- Case Studies and Real-World Examples and Practical Applications and Projects
- Capstone Project.

Course Outcomes (COs):

At the end of the course, the student should be able to:

1. Understand the capabilities of PySpark for handling and processing large datasets. (PO-1,2,3, PSO-2)
2. Design a variety of visualizations and interactive dashboards using Tableau. (PO- 1, 2,3, PSO-3)
3. Implement an end-to-end data visualization application by Using PySpark to efficiently handle, clean, and transform large datasets (PO- 1,2,3,5-PSO-3)

Course Assessment and Evaluation:

Continuous Internal Evaluation (CIE) : 50 Marks		
Assessment Tools	**Marks**	**Course Outcomes addressed**
Internal Test-I (CIE-I)	20	CO1, CO2, CO3
Regularity/ Continuous assessment	10	CO1, CO2, CO3
Average of the two CIE will be taken for 30 marks		
Other Components		
Project Implementation	20	CO1, CO2, CO3
The Final CIE out of 50 Marks = CIE TEST+ Continuous assessment +Project Implementation		
Semester End Examination (SEE)		
Course End Examination	50	CO1, CO2, CO3

PROJECT WORK	
Course Code: CSP81	**Credits: 0:0:12**
Prerequisite: NIL	**Contact Hours:**
Course Coordinator: Dr. S. Rajarajeswari	

Course Contents

As a part of project, all the eligible final year students must carry out the following activities:

1. Students should form a group to carry out their project. The minimum group size is 2 and maximum group size is 4.
2. The groups will be attached to one Internal Guide (and Co-guide if necessary) by the Department.
3. Students can carry out their project in-house or in a reputed organization (to be approved by Internal Guide and HOD).
4. Identify the problem statement based on the current state of Art and trends in the area of interest.
5. Based on the survey, identify the project requirements and do feasibility study.
6. Identify and draw a system level architecture by showing subsystems and their input/output need.
7. Implement the programs using step by step for each module.
8. Integrate and examine the implementation and test the project scope and the requirements.
9. Prepare Project document and the demonstrating their work.
10. Publish the Project work in a Scopus indexed Conference/Journal with quartile ranking (Q1, Q2, Q3)
11. The Continuous Internal Evaluation is based on a presentation and report for 100 marks scaled down to 50 marks.
    - The evaluation will be done by the internal guide and a co- examiner twice during the semester.
    - Weekly report to the Guide (10 marks)
    - Mid-semester evaluation: Students must do a group presentation and produce documents of system requirements, and system design (during 6^{th} week) (40 marks)
    - Final Evaluation: At the End of the semester students must do a group presentation, demonstrate the project work and submit the complete report. (during 13th week) (50 marks)
12. The SEE Project Viva Voce will be conducted for 50 marks with an Internal and External Examiner.

Evaluation:

The rubrics for project evaluation are provided as per the below tables.

Criteria for Evaluation	**Level A (10) 90-100**	**Level B (8) 75-90**	**Level C (6) 50-75**	**Level D (5) Up to 50**
Data Elicitation Phase	Has investigated new trends in their area of interest, Review the challenges in that area. Data elicitation should include new concepts.	Has investigated new trends in their area of interest, Review some challenges in that area	Has investigated new trends in their area of interest.	Has not investigated much on new trends in their area of interest
Problem Definition	Has investigated problem domain extensively	Problem domain well understood, clear and specific description of problem, relevance well identified	Moderately awareness of problem domain, clear description, broad idea about relevance to current technical and social context	Minimal awareness of problem domain, Vague description, little idea about relevance to current Technical and social context
Planning	Precise Schedule and Effort Estimation using tools.	Precise Schedule and Effort Estimation Manually.	Schedule and Effort Estimation.	Inappropriate Schedule and Effort Estimation.
Project management	Has taken leadership role in the project and monitored progress of the project. In addition, has completed all tasks assigned to him.	Has monitored the progress of the project and completed all tasks assigned to him.	Has completed all the tasks assigned to him.	Has not completed all the tasks assigned to him.
Literature Survey	Has read more than 10 papers from reputed journals or 15 papers from conference and 3 books in the area of the project.	Has read more than 7 papers from reputed journals or 12 papers from conference and 2 books in the area of the project.	Has read more Than 5 papers from reputed journals or 8 papers from conference and 1 books in the area of the project.	Minimal, mostly from general sources, without focused study.
Requirements Specification	Complete functional, Non-functional, Performance, Security related, Clear and Measurable (in terms of SMART Matrix	Partial functional, Non-functional, Performance, Security related, Clear and Measurable (in terms of Smart Matrix	In-Complete functional, Non-functional, Performance, Security related, Clear and Measurable (in terms of SMART Matrix)	Few, narrow and incomplete requirements

System Design	The team has to play a role of main Architect in the project , Designed all the components of the project	The team has to play the role of main Architect in the project , Designed 90% of the components of the project	The team has to play the role of main Architect in the project , Designed 75% of the components of the project	The team has to play the role of main Architect in the project, Designed 50% of the components of the project
Implementation	Has decided the relevant tools and platforms required for the project by evaluating alternatives. coded all the components designed by him by following the standard coding guidelines.	Has coded all the components designed by him by following the Standard coding guidelines.	Has coded all the Components designed by him by not following the standard coding guidelines.	Has not coded all the components designed by him by not following the standard coding guidelines.
Testing and Results	Meets all the requirements, Optimized Solution, Proper Test Plan, Has performed Integration Testing, Performance Testing	Barely meets all the requirements, Not Optimized Solution, Poor Test Plan, Has not performed Integration Testing, Performance Testing	Barely meets all the requirements, Not Optimized Solution, Poor Test Plan, Has not Performed Integration Testing, Performance Testing	Haphazard testing,barely meets requirements, unable to infer results.
Report Writing	Excellent Organization,	Good Organization, No technical or Grammar errors, Concise and Precise, Incomplete documentation, done on Latex	Average Organization, No technical or Grammar errors, Concise and Precise, Incomplete documentation, Not done on Latex	Poor Clarity in technical contents and organization, error in grammar, not done in Latex
Presentation and Viva-voce	Excellent Professional and Technical communication, Effective Presentations, able to analyze technically and clarify views in viva-voce	Good Professional and Technical communication , Effective Presentations.	Average Professional and Technical communication, Effective Presentations, Unable to analyze technically and Clarify views in viva-voce	Poor Technical communication, Not an Effective Presentations, Unable to analyze technically and clarify views in viva-voce

Rubrics for Final year Project Paper (IEEE Format)			
Criteria and Qualities	**Poor (1-4)**	**Adequate (5-7)**	**Good (8-10)**
Abstract (1)			
Content & Length	The abstract does not cover each sectionof the paper; conclusion do not match study findings The abstract exceeds the word limit (250words)	The abstract covers most sections of the paper The abstract is an appropriate length	The abstract covers all sections of the paper and coveys the key findings from the study in a clear way The abstract is an appropriatelength
Introduction (3)			
Literature Review	Literature reviewed has weak or no connection to the topic under study. A clear rationale for the study aim/ purpose (the gap in the literature) is notidentified. Major sections of pertinent literature are omitted or literature reviewed is not from scholarly sources	Literature reviewed relatesto the study topic. All major sections of the pertinent literature are included, but may not be covered in depth or a few areas of pertinent literature are not covered. Most articles/ sources reviewed are from scholarly sources but a few are not.	Literature reviewed relates to themain topic and sets up the rationale for the study aim/ purpose by clearly identifying a gap in the literature. Introduction covers relevant andcurrent articles in detail. Only scholarly articles are used to build the argument for the need for the study.
tudy Aim/ Purpose	Neither implicit nor explicit reference is made to the study aim or purpose or the study aim appears unrelated to theliterature reviewed.	Readers are aware of the overall problem or aim of the study but the aim is not clear or the study aim is not clearly linked to a gap in the literature	The aim clearly flows from the groundwork laid/ literature reviewed and a clear case is madefor the need for the study aims based on a gap in the literature.
Method	Critical details necessary to understandhow the study was conducted are lacking. Sections such as study design,procedures, measures/ instruments, analytic approach are missing.	Study methods are generally described but information regarding nuances of how the study was conducted is more limited.	The study design is described indetail. If quantitative data were collected, psychometric properties of instruments are provided (e.g. validity and reliability) are provided. The sampling strategy (inclusion/ exclusion criteria), study settingand data collection procedures are described in detail. The approach used for data analysesare described.

Results (2)			
Presentation of Data/ Findings	Results are poorly Described and/or do not align withdescription of study methods. The Results section includes "conclusions"instead of simply presenting the data and/ or analyses.	Results are adequately described and aligned withdescription of study methods. Analyses of dataare completed correctly.	Results/products/outcomes are described in detail and align withdescription of study methods. Analyses of data are sophisticatedand precise
Tables and Figures	Tables/Figures are missing and/or unlabeled. If present, they do not clearly present the study findings/data and/ or are redundant with one another.	Tables/Figures are present in the paper and are labeled. Tables/ Figures adequately present data/ findings but may be redundant with data presented in the text	Tables/Figures are present and are labeled. Tables and Figuresprovide critical information and are organized in such a way as toenhance understanding of the study results.

Course Outcomes (COs):

At the end of the course, the students should be able to:

1. Review the current state of Art and trends in their area of interest and identify a suitable problem in their chosen subject domain with justification. (PO-1,2,6, 7, 9, 10, 11, PSO-2,3)
2. Survey the available research literature/documents for the tools and techniques to be used. (PO-1, 2, 5, 8, 9, 10, 11, 12, PSO-2,3)
3. Examine the functional, non-functional, and performance requirements of their chosen problem definition. (PO-1,2,4, 9, 10, 11, 12, PSO-2,3)
4. Design system architecture and different components and develop all the system components using appropriate tools and techniques. (PO-1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, PSO-2,3)
5. Work effectively in a team and use good project management practices and defend the project work as a team (PO-5, 8, 9, 10, 11, 12,PSO-2,3)

RESEARCH / INDUSTRIAL INTERNSHIP	
Course Code: INT82	**Credits: 0:0:5**
Prerequisite: NIL	**Contact Hours:**
Course Coordinator: Dr. S. Rajarajeswari	

Course Contents

Guidelines:

- The student can do the Internship during the summer vacation after the 2nd,4th,6th semester and in 8th semester.
- The student should take prior permission from the department committee before carrying out the internship. The offer letter should be submitted to the Dept. coordinator.
- The duration of the Internship is 3- 4 -6 weeks after each semester (total Six months).
- The Internship assessment form assessed by the company mentor, report of the Internship, Internship completion certificate, needs to be submitted to the department coordinator after completion of the Internship.
- The department will constitute a committee for the evaluation of Internship of student and based on the following rubrics, the student will be evaluated.

Internship Assessment – Rubrics

Each supervisor must fill a rubric for each student:

Tools and	Basic	Good	Very Good (10 Pts)	Total
new Technology Learnt	**(0-4 Pts)** Few sources at the Industry, aware of quality of resources and relevance to tools and Techniques at hand	**(5-7 Pts)** Multiple sources of high quality, good judgment of the information, identification of gaps in knowledge at the Industry and Academics.	Multiple sources of high quality, well researched and analyzed, continuous efforts at acquiring Information. Identification of the application of the tools and Technology learnt to the present market.	
Relevance of the topic chosen to the current market	Fairly Relevant	Moderately Relevant	Highly Relevant	

Rubrics for the Report Writing, Demo and Presentation are maintained in a separate workbook

Course Outcomes (COs):

At the end of the course, students will be able to

1. Schedule milestones of deliverables and formulate the requirements of the proposed work. (PO-2,9,11, PSO-1,2)
2. Apply the engineering knowledge to develop software in an industry setting. (PO-1,2,3,5, PSO-1,2)
3. Develop the inter-personal skills required to work in a professional team. (PO-9, 10, 11, PSO-2, 3)
4. Engage in independent study of technology required for development of software. (PO-12, PSO-2, 3)
5. Demonstrate the project and appraise its effectiveness. (PO-10, PSO-3)